Lämmitysjärjestelmän datankeruu ja analysointi

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulututkinnon opinnäytetyö

Valkeakoski, Sähkö- ja automaatiotekniikan koulutusohjelma

Syksy, 2017

Atte Partanen

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Koulutus Sähkö- ja automaatiotekniikan koulutusohjelma
Kampus Valkeakoski

Tekijä Atte Partanen **Vuosi** 2017

Työn nimi Lämmitysjärjestelmän datankeruu ja analysointi

Työn ohjaaja/t Timo Väisänen

TIIVISTELMÄ

Tässä opinnäytetyössä käytiin läpi esineiden internetin perusteita ja kehitettiin sovellus prosessiarvojen tallentamiseen pilvipalveluun. Sovelluksen on tarkoitus tulla Pirkanmaan liiton rahoittamaan Tarkalla ohjauksella energiatehokkuutta -hankkeeseen. Työssä perehdyttiin nykyisiin viestintäprotokolliin ja niiden toimintaan esineiden internet sovelluksessa. Tämän lisäksi työssä käydään eri tietokantavaihtoehtoja ja niiden hyödyntämistä pilvipalveluissa. Työn tavoitteena oli luoda sovellus, jossa hyödynnetään pilvipalvelun esineiden internet sovelluksia.

Työn aikana kerättiin tietoa erilaisista esineiden internetissä käytettävistä viestintäprotokollista ja tietokantavaihtoehdoista. Työssä myös tutustuttiin erilaisiin pilvipalveluntarjoajiin, joista suurimmassa keskiössä oli Microsoft Azuren -pilvipalvelun konfigurointi ja sinne lähetettävän datan keräys tietokantaan.

Työssä kehitettiin sovellus, jossa lähetettiin dataa Microsoft Azuren -pilvipalveluun, jossa palvelussa sisäisesti data säilöttiin pilvipalvelun tietokantaan. Sovelluksen data lähettämiseen ja lukemiseen käytettiin hyödyksi Node-RED kehitysympäristöä, jonka avulla pystyttiin muodostamaan informaatiosivu, jossa luetusta datasta muodostettiin kuvaaja.

Avainsanat IoT, Microsoft Azure, Node-RED, pilvipalvelu

Sivut 37 sivua, joista liitteitä 5 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Electrical and Automation Engineering
Valkeakoski

Author Atte Partanen **Year** 2017

Subject Data collection and analysis of a heating system

Supervisors Timo Väisänen

ABSTRACT

In this Bachelor's thesis project, the author examined the concept of The Internet of Things (IoT), and developed an application to storage process values to the cloud service. The application was produced for the project "Energy efficiency with precise control" funded by the Council of Tampere Region. This thesis goes through current messaging protocols and how to use protocols in the Internet of Things application. After that the focus moves to examining different kind of databases and uses in cloud services. The Main goal of this project was to create an application to use cloud service as an IoT platform.

During the project information was collected on different options for messaging protocols and databases to communicate in an IoT environment. One goal was also to go through different types of cloud services. The Focus of this thesis was configuring Microsoft Azure Cloud Service for data collection and storage in a database.

In the thesis, an application was developed to send data to the Microsoft Azure Cloud Service, where the data was internally stored in a cloud service database. The Node-RED development platform was used to send, and query data and simultaneously to make a plot for the collected data.

Keywords Cloud service, IoT, Microsoft Azure, Node-RED.

Pages 37 pages including appendices 5 pages

SISÄLLYS

LYHENTEET JA TERMIT

Asynkroninen	Tarkoittaa ei-reaaliaikaista kommunikaatiota, eli osapuolet eivät ole riippuvaisia toistensa ajasta ja paikasta.
Broker	Välittäjä, joka jakaa tiedot julkaisijalta tilaajalle.
Endpoint	Viestintäpääte, joka keskustelee sovelluksen tai laitteen välillä.
GB	Yleisesti tietoliikenteessä käytetty lyhenne gigatavusta.
IoT	Internet of Things, Esineiden internet.
IP	Numerosarja, jota käytetään verkkosovittimien yksilöintiin.
JSON	JavaScript Object Notation, Tiedosto muoto, jota käytetään yleisesti ohjelmoinnissa ja tiedonvälityksessä.
M2M	Machine to Machine, tarkoittaa laitteiden välistä kommunikaatiota.
NoSQL	Not only SQL, tietokanta, jolla ei ole määrättyä mallia.
OSI-malli	7 kerroksinen malli, joka standardoi tietoliikennetoimintoja.
Query	Tarkoittaa kyselyä, jonka avulla tietokannalta voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä.
RU	Request Units määrittää, kuinka monta kertaa tietoa voidaan lukea kokoelmasta.
SD	Secure Digital, tarkoittaa muistikorttistandardia.
SQL	SQL on standardoitu kieli, jota käytetään tietokannoissa.
Tietokanta	Kanta, johon tietoa on kerätty yhteenkuuluvien tietojen joukkoina.
USB	Universal Serial Bus on sarjaväyläarkkitehtuuri, jota käytetään oheislaitteiden liittämiseen.
WLAN	Langaton lähiverkkotekniikka.
XML	Extensible Markup Language. Käytetään tiedon esittäminen ihmisen ja koneen ymmärrettävässä muodossa.

1 JOHDANTO

Automaatiojärjestelmissä käytettävät varastointitavat ja viestintäprotokollat ovat kehittymässä kovaa tahtia. Enää ei ole välttämätöntä kuluttaa resursseja omaan palvelimeen, joka kerää dataa tietokantaan ja vastaa myös datan käsittelystä. Datan keräämisen avuksi voidaan ottaa käyttöön pilvipalveluita, joiden avulla voidaan datankeräys ja -käsittely hoitaa jossain muualla kuin käytössä olevassa järjestelmässä, jolloin järjestelmä kustannukset voivat olla pienemmät. Pilvipalveluiden avulla voidaan tulevaisuudessa tietokannan kokoa laajentaa ilman järjestelmän päivitystä, jolloin järjestelmään saataisiin lisää tallennustilaa.

Opinnäytetyön tavoitteena on tutkia automaatiossa käytettyjä tapoja lähettää prosessista mitattuja arvoja suoraan pilvipalvelimelle. Työssä tutkitaan tämän hetkisiä pilvipalvelun tarjoajia ja niiden tarjoamia palveluita esineiden internet sovelluksien toteuttamiseen. Työssä on tarkoituksena tutkia erilaisia esineiden internetissä käytettäviä viestintäprotokollia ja käytössä olevia tietokantoja. Työssä tarkoituksena on kehittää sovellus, jolla voidaan tietoa lähettää pilvipalvelimelle, jossa data tallennetaan tietokantaan pilvipalvelussa. Sovelluksen on myös tarkoitus lukea dataa pilvipalvelimen tietokannasta.

Aihe on ajankohtainen, koska automaatiossa ollaan menossa koko ajan enemmän siihen suuntaan, että kaikista yksinkertaisimmista toimilaitteista on mahdollista saada tilatietoja, joita voidaan lukea ja monet laitevalmistajat tarjoavat omia pilvipalvelimia laitteiden tietojen tallentamiseksi. Kun dataa kerätään talteen, on sille tarpeen kehittää jonkinlainen sovellus, jolla dataa voidaan esittää käyttäjälle tarpeellisena ja ymmärrettävässä muodossa.

2 TYÖN KUVAUS

2.1 Yleistiedot

Tarkalla ohjauksella energiatehokkuutta -hanke (TOE-hanke) on Pirkanmaan liiton rahoittama hanke, jonka tavoitteena on kehittää ja rakentaa energiatehokas ja energiaa säästävä hybridilämmitysjärjestelmä. Projekti kehittäminen aloitettiin keväällä 2016 HAMKin Valkeakosken sähkö- ja automaatiokoulutusohjelman kanssa. Hankkeen tavoitteena on lämmitysjärjestelmän automaatio-ohjauksen kehittäminen, siten että tarvittava lämmitysenergia saataisiin tuotettua mahdollisimman energiatehokkaasti ja mahdollisimman edullisesti.

2.2 Lämmitysjärjestelmä

Lämmitysjärjestelmä on suunniteltu siten, että sitä olisi mahdollista siirtää tarpeen tullen eri paikkaan ja mahdollisuus käyttää kohteessa, jossa lämmitysenergiaa tarvitaan. Lämmitysjärjestelmä koostuu lämmönlähteistä, joita ovat lämmityskattila ja aurinkokeräimet. Lämmityskattilaa voidaan lämmittää hakkeella tai muulla kuivalla aineella. Kattilaa voidaan myös lämmittää bioöljyllä ja sähköllä. Lämmitysjärjestelmän lämmönvarastointiin käytetään kolmea identtistä vesivaraajaa, joita voidaan erikseen varata ja purkaa. Kahteen vesivaraajaan tulee faasimuutosmateriaalia, jota on tarkoitus verrata vesivaraajaan, jossa on pelkästään vettä. Hankkeessa on tarkoituksena tutkia, voidaanko faasimuutosmateriaaleja pitää tulevaisuuden lämmönvarastoinnin vaihtoehtona.

Lämmitysjärjestelmän suuressa osassa on automaatio, joka vastaa kaikista ohjauksesta ja mittauksista, joita järjestelmässä tehdään. Kun järjestelmä seuraa kaikkea mitä prosessissa tapahtuu, on siis automaatiojärjestelmän pystyttävä reaaliaikaiseen datankeruuseen. Suurien datamassojen vuoksi monesti automaatiojärjestelmä joutuu keräämään ja käsittelemään dataa. Tähän voidaan ottaa avuksi pilvipalvelu, johon järjestelmän mittaamat tiedot lähetetään ja missä ne käsitellä. Silloin vältytään suurilta tiedostokuormilta järjestelmän puolella. Kaikki tietojen käsittelyyn vaadittu prosessointi hoidetaan palvelimella, jolloin säästytään vaativilta prosessoineilta automaatiojärjestelmässä.

3 TEOLLINEN INTERNET

3.1 Määrittely

Käsite teollinen internet on usein mielletty tarkoittamaan vain perinteistä teollisuuden hyödyntämää internetiä. Teollisen internetin hyödyt eivät rajoitu laitetoimittajan hyötyihin, vaan hyödyt tulevat siitä, että koneet ja laitteet eri laitevalmistajilta voivat toimia samassa ympäristössä, joka helpottaa hallitsemaan tuotantoprosessia tehokkaammin. (Collin & Saarelainen 2016, 30.)

3.2 Teollisuus 4.0

Teollisuus 4.0 on tunnetuin strateginen muutosohjelma teollisen internetin hyödyntämistä uudessa automaatioteknologiassa. Se on Saksan valtion vuodesta 2012 ajama kansallinen hanke, jonka lähtökohtana on maan valmistavan teollisuuden kilpailukyvyn säilyttäminen ja vahventaminen. (Collin & Saarelainen 2016, 37.)

Teollisuus 4.0 on teollisuusautomaation uusi sukupolvi, joka keskittyy suuremmaksi osaksi digitalisaatioon ja analytiikkaan. Keskiössä ovat tuotteen valmistus ja tehdastuotannon liittäminen internetiin. Uudet ratkaisut yhdistävät ihmiset, tuotteet ja palvelut, joiden avulla luodaan autonomisia kokonaisuuksia, joissa koneet, tuotantoprosessit ja varastot keskustelevat keskenään reaaliaikaisesti. Tämä mahdollistaa paremman tuotteen elinkaaren ja toimitusketjun hallinnan. (Collin & Saarelainen 2016, 37.)

3.3 Esineiden internet

Esineiden internet eli IoT on nykyään kuuma puheenaihe työpaikoilla kuten myös ihan arkielämässä, mutta mitä tämä käsite tarkoittaa. Esineiden internet tuli esille ensimmäisen kerran, kun vuonna 1980 Carnegien Melon yliopisto opiskelijat kehittivät sovelluksen, tarkistamaan onko virvoitusjuoma-automaatissa juomia jäljellä. Esineiden internet käsitteellä tarkoitetaan sitä, että laitteet pystyvät lähettämään tai vastaanottamaan tietoa. (IoT-Agenda 2016.)

Esineiden internet termiä käytetään kuvaamaan valtavaa verkkoa, erilaisia esineitä, jotka ovat ennen olleet yksitilaisia laitteita. Tämä sisältää kaiken älypuhelimista kahvinkeittimiin, pesukoneisiin, lamppuihin ja melkein mihin tahansa muuhun, minkä voi liittää verkkoon. (Morgan 2014.)

Esineiden internet tarkoittaa nykypäivänä enemmän kuin älykkäät kodit ja niihin liitetyt sovellukset. Tulevaisuudessa voidaankin odottaa, että mukaan tulevat älykkäät kaupungit, jossa liikennevalot ja -merkit mittaavat liikenteen kulkua ja ohjaavat mittauksien perusteella liikennettä. (Kobie 2015.)

Aloja, jotka hyödyntävät IoT-ratkaisuja, ovat prosessit, joiden tavoitteena on paras mahdollinen hyötysuhde. Tällaisia ovat hissit, keittimet, pakastimet ja ilmastointilaitteet. Myös sairaaloissa voidaan monitoroida potilaiden terveyttä puettavilla antureilla. IoT-teknologian avulla voidaan helpottaa henkilökuntaa kunnossapidon operaatiossa. (Hicklin, Shurvinton & Beard 2015, 10-11.)

Esineiden internetin väite ”jokaiselle laitteelle on oma IP-osoite” ei nykyisten IoT-laitteiden kanssa pidä paikkaansa. Laitteen kykyyn kommunikoida ei välttämättä vaadita internet yhteyttä, jolloin laitteella ei ole IP-osoitetta, vaan nykyisillä langattomilla teknologioilla pystytään laitteet saamaan keskustelemaan keskenään. Yhtenä esimerkkinä on EnOcean-sensorit, jotka ovat langattomia IoT-laitteita, jotka keräävät tarvittavan energian ympäristöstä. Esimerkiksi kytkimen painalluksen aiheuttama kineettinen energia riittää sensorin datapaketin lähettämiseen.

3.4 **Turvallisuus**

Teollisen internetin haasteena on tietoturva, joka saattaa usein jäädä ilman huomiota yrityksen johdossa. Teollinen internet ei poikkea muusta internetistä laisinkaan, vaan data liikkuu samoja kaapeleita pitkin, kun muukin internetliikenne, jolloin järjestelmää suunnitellessa on otettava huomioon, että data liikkuu lähetyspisteestä loppupisteeseen turvattuna. (Vieno 2015.)

Suurin osa kaikista verkkoon kuuluvista laitteista kerää paljon henkilökohtaista tietoa, johon käyttäjät eivät halua muiden pääsevän käsiksi. Esimerkiksi jotkin laitteet tietävät, milloin olet kotona tai mitä elektroniikkaa käytät. Nämä tiedot jaetaan usein toisten laitteiden kanssa ja tallennetaan tietokantoihin. Tällöin arkaluotoiset tiedot voivat olla kyberrikollisille suurta valuuttaa. (Kobie 2015.)

Käyttäjän ei pitäisi luottaa siihen, että laitetoimittajat olisivat hoitaneet tietoturvansa hyvin. Yleensä laitetoimittajat tuottavat vain laitteen ja sovelluksen, ja käyttäjä toteuttaa sitten käytettävän sovelluksen ja joutuu itse huolehtimaan, että laite on turvallisesti kytkettynä verkkoon. (Uusitalo 2016.)

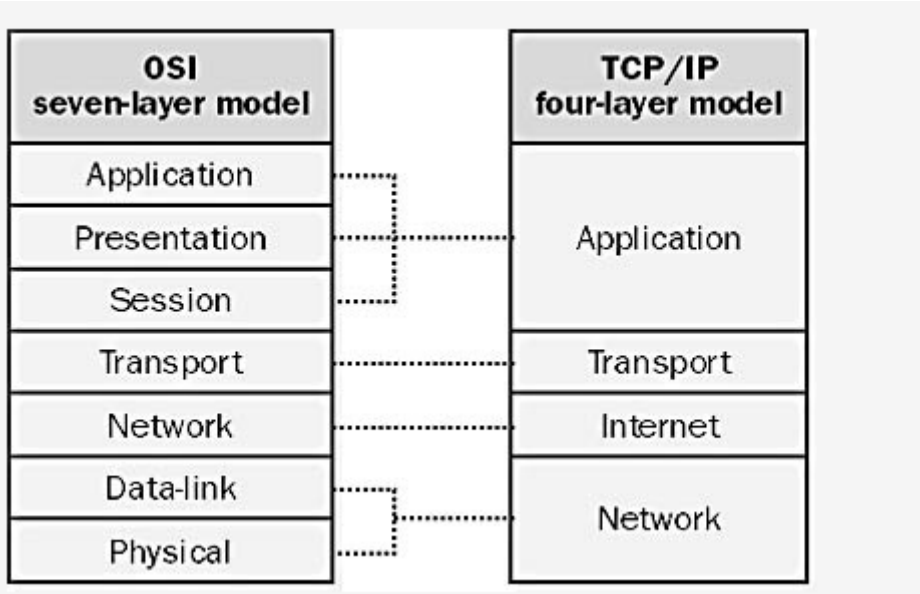
Laitteiden tietoturvallisesta kytkemisestä ovat vastuussa käyttäjä ja myös arkkitehtuurin suunnittelija ja toteuttaja. Arkkitehtuurin toteuttama verkko tulee olla suunniteltu siten, ettei kukaan ulkopuolinen pääse laitteeseen käsiksi. Vaikka nämä seikat otettaisiin huomioon, voi olla mahdollista, että laitetoimittajat ohjeistavat käyttäjiä ohittamaan tietoturvallisuutta vahventavia tekijöitä. (Uusitalo 2016.)

Kaiken kaikkiaan esineiden internet on suhteellisen turvallinen. Hyökkäyksen kohteeksi joutuminen on epätodennäköistä ja hyökkäykset eivät todennäköisesti tee yhtään enempää vahinkoa kuin esimerkiksi tavalliset hyökkäykset kotitietokoneeseen. Tämä voi muuttua tulevaisuudessa, kun esineiden internet ja kodin automaatio yleistyvät. Hyökkäysten toteutuminen kohteisiin on kannattavampaa, jos tietoturvan kehittämiseen ei panosteta tarpeeksi. (Kobie 2015.)

4 PROTOKOLLAT JA TIETOKANNAT

4.1 Viestintäprotokolla

Esineiden internetin viestintäprotokollat käyttävät liikennöintiin TCP/IP-verkkoa, jonka suhteen OSI-malliin voidaan nähdä kuvassa 1. TCP/IP-protokolla jaetaan 4 eri luokaan. Eri tasot määräävät, minkälaiseen verkkoon laitteet ovat kytkettynä ja kuinka viestintäprotokolla hoitaa keskusteluverkoston rakenteen.



Kuva 1. TCP/IP-protokollakerrosten suhde OSI-malliin (TCP-IP n.d.)

Sovelluskerros määrittelee, kuinka sovellus hyödyntää kuljetuskerroksen protokollia lähettämään datan kohteeseen. Jokaisessa kerroksessa on monia erilaisia protokollavaihtoehtoja hoitamaan tehtäviä, joita eri kerroksissa tarvitaan suorittaa. (TCP-IP n.d.)

Kuljetuskerroksen tarkoituksena on muodostaa yhteys kahden isännän väliseen tiedonsiirtoon. Kuljetuskerros vastaanottaa tietoa sovelluskerrokselta. (TCP-IP n.d.)

Verkostokerroksen pääasiallinen tarkoitus on järjestää tai käsitellä tietojen siirtoa verkossa. Tietojen siirtämisellä yleensä tarkoitetaan tiedon reitittämistä paikallisverkon yli, johon pääsääntöisesti käytetty protokolla on IP. (TCP-IP n.d.)

Siirtokerrosta ja fyysistä kerrosta käytetään verkkoliitäntään. Kerros koostuu laiteajureista ja järjestelmään liitetyistä verkkoliitännäiskorteista, jotka huolehtivat viestinnän yksityiskohdista. (TCP-IP n.d.)

IoT-laitteiden viestintäprotokollan yleisimpiä käytössä olevia viestintä tapoja ovat julkaisija/tilaaja-malli ja pyyntö/vastaus-malli. Näistä yleisin käytössä oleva malli on julkaisija/tilaaja-malli, jossa broker kerää tietoa siitä, kuka julkaisee dataa ja kuka dataa haluaa nähdä. Julkaisija ei välttämättä tiedä, kuka dataa haluaa lukea, riippuen käytössä olevasta protokollasta. Tärkeydessä on julkaisijan aihe, mihin tietoa lähetetään, tämä on elintärkeää tilaajalle, joka lukee tietoa aiheen perusteella.

Toinen viestintään käytetty tapa on pyyntö/vastaus-malli, jota yleisesti käytetään HTTP-ympäristössä. Malli toimii siten, että toinen osapuoli lähettää pyynnön, jossa toinen osapuoli vastaa pyynnön mukaisesti.

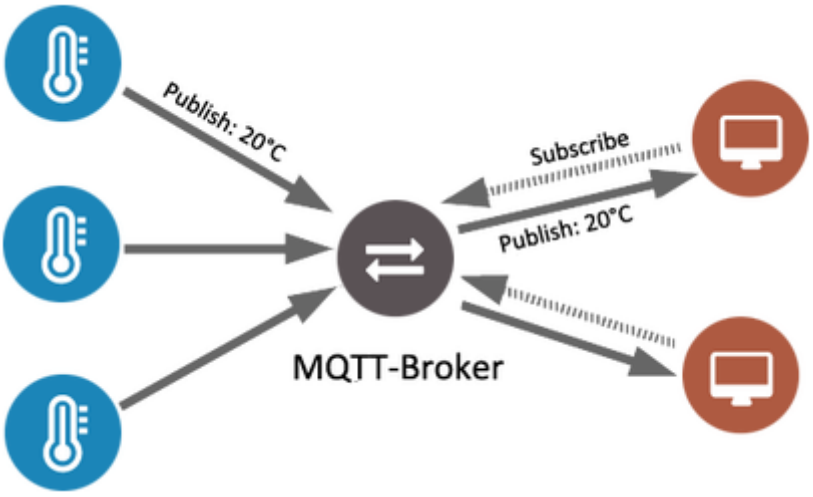
Viestintäprotokollat ovat IoT-laitteiden käyttämiä. Ne mahdollistavat laitteiden lähettämän tietojen turvaamisen ja tietoliikenneverkkoon yhdistämisen. Protokollat määrittävät viestinnän nopeuden, tehokkuuden ja turvallisuuden.

4.1.1 MQTT

MQTT (Message Queue Telemetry Transport) on IoT-liitännäisprotokolla, joka on suunniteltu erittäin kevyeksi viestintävälineeksi. Se on hyödyllinen rajatuilla yhteyksillä, kuten kotiin liittyvässä automaatiossa ja pienissä laitteissa. Se soveltuu hyvin myös käytettäväksi mobiilisovelluksille pienen pakettienkoon ja virrankulutuksen vuoksi. (MQTT n.d.)

MQTT-protokolla soveltuu datan keräämiseen suuressa mittakaavassa eri mittauksia, jotka kerätään myöhemmin keskitettyyn it-järjestelmään, tyypillisesti pilvipalvelimelle. Protokolla ei sovi kenttälaitteiden keskeiseen kommunikointiin eikä suoraan viestintään eri vastaanottajille. Vaan protokollan tavoitteena on pitää laitteelta lähtevä viesti mahdollisimman puhtaana. (Collin & Saarelainen 2016, 187.)

Protokollan suuria vahvuuksia on, että sen viestintärakenne on julkaisija/tilaaja-mallinen. Julkaisijan tarkoituksena on lähettää tiedot MQTT-broker rajapinnalle, joka harjoittaa viestit lähettämällä tiedon eteenpäin siitä kiinnostuneille tilaajalle, kuten kuvassa 2. Julkaisijaa ei kiinnosta, kuinka monta laitetta viestiä lukee. Se ei myöskään kerää tietoa, kuinka monta laitetta on yhteydessä MQTT-verkossa. Tämän takia se onkin yksi yleisimmistä käytössä olevista IoT-laitteiden viestintäprotokollista.



Kuva 2. MQTT-välittäjän toiminta (MQTT-Broker n.d.)

MQTT-protokollan avulla voi yhdellä laitteella olla monta eri aihetta, mihin julkaistaan tietoa. Automaatiojärjestelmän turvallisuus on aika varma, koska laitteet eivät ole riippuvaisia toisistaan, vaan julkaisija ja tilaaja voivat olla eripuolella maailmaa. Protokolla myös tarjoaa mahdollisuuden viimeisimmän arvon historiatiedon tallentamiseen. Tämän avulla voidaan taata, että tieto välittyy aihetta tilanneelle laitteelle, vaikka julkaisija ei ole lähettänyt tietoa brokerille.

4.1.2 AMQP

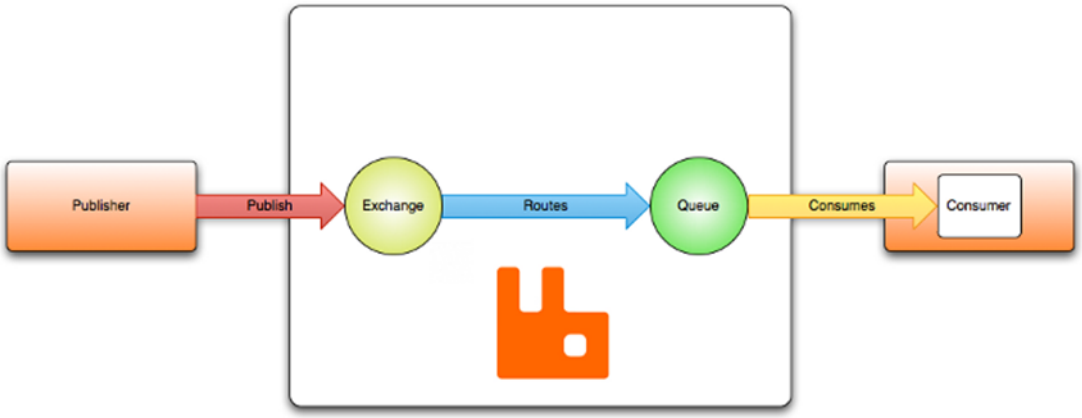
AMQP (Advanced Messaging Queing Protocol) on avoin, yrityspuolelle suunnattu viestintäprotokolla, jolla voi lähettää viestejä sovellusten tai organisaatioiden välillä. Se yhdistää järjestelmät ja avustaa liiketoimintaprosesseja tarvittavalla tiedolla, jonka jälkeen se lähetetään tietoa luotettavasti eteenpäin. Luotettavuus onkin yksi avaintekijä, miksi erityisesti yritykset suosivat AMQP-protokollaa. (AMQP n.d.)

AMQP-protokolla kykenee lähettämään suuria määriä pienikokoisia viestejä nopealla tahdilla. Se on suunniteltu toimimaan eri toimittajien järjestelmissä, joten protokolla soveltuu suorituskykyä vaativiin liikennöintiin eri palvelimilla. AMQP-protokollaa käyttävät suuret yritykset, kuten pankit eri puolella maailmaa, Nasa ja Google. (Collin & Saarelainen 2016, 188.)

AMQP-välittäjä toimii siten, että viestit tulevat julkaisijalta, jonka viestit reititetään kuluttajalle. Välittäjän viestit on myös mahdollista tallentaa viestijonoihin ja lähettää valmis kokoelma asiakkaalle. Verkkoprotokollan takia julkaisija, kuluttaja ja välittäjä voivat olla eri laitteella. (Rabbitmq n.d.)

AMQP-välittäjä voi joko lähettää tiedon kuluttajalle tai kuluttaja voi olla asemassa, jossa se itse pyytää tietoa välittäjältä kuvan 3 mukaisesti. Ver-

koston rakenne toimii siten, että kun viesti on lähetetty kuluttajalle kuluttaja vastaa välittäjälle, että viesti on saapunut perille automaattisesti tai vaihtoehtoisesti vasta sitten kun sovelluksen kehittäjä on näin päättänyt tehdä. (Rabbitmq n.d.)

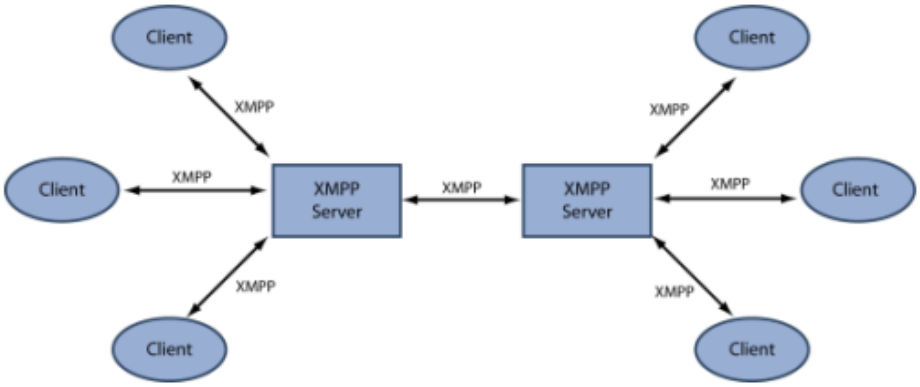


Kuva 3. AMQP-liikkennöinti (Rabbitmq n.d.)

4.1.3 XMPP

XMPP (Extensible Messaging and Presence Protocol) on avoin, XML-pohjainen viestintäprotokolla, jonka tarjoaa mahdollisuuden muun muassa pikaviestintään, ryhmäkeskusteluihin ja ääni- ja videopuheluihin. Alun perin se julkaistiin 1999 nimellä Jabber, ja sen tarkoituksena oli silloin tarjota vaihtoehto ihmisten pikaviestikeskusteluihin, joita käytettiin lähinnä Microsoft Networksillä. (XMPP n.d.)

XMPP-protokolla sopii viestintään, jossa datan on tarkoituksena kulkea pitkälle monen pisteen läpi. Tämän takia XMPP-protokollaa käytetään yleisesti teollisessa ja esineiden internetissä, esimerkiksi smart grid -ratkaisuissa, jotka koostuvat älykkäistä etäluettavista sähkömittareista. Protokollan vahvuutena on helppo, sähköpostin kaltainen laitteen osoittaminen. Esimerkiksi osoittamiseen voidaan käyttää rakennetta nimi@domain.com, tämä auttaa monien eri laitteiden yhdistämistä. Protokolla on myös tietoturvallinen ja helposti skaalautuva, mutta heikkoutena protokollalla on, ettei sitä ole suunniteltu hirveän nopeaksi, joten protokolla käyttötarkoitukset rajoittuvat. (Collin & Saarelainen 2016, 188.)



Kuva 4. XMPP-viestintämalli (Isode n.d.)

XMPP-protokollan viestintä toimii kuten sähköposti. Jokaiselle asiakkaalle on annettu oma sähköpostiosoite, joka vastaa XMPP-palvelimelle asetettua verkkotunnusta. XMPP-protokolla mahdollistaa myös palvelimien välisen kommunikaation kuten kuva 4 osoittaa. (Isode n.d.)

4.1.4 HTTP

HTTP (Hypertext Transfer protocol) käytetään yleisesti selainten ja www-palvelimien liikennöintiin. Protokolla perustuu siihen, että selain avaa TCP-yhteyden palvelimelle ja lähettää pyynnön. Palvelin vastaa lähettämällä sopivan vastauksen halutussa muodossa. (Karasiewicz 2013.)

HTTP-protokolla on, heikko viestintäprotokolla ajattelen IoT-laiteita. Laitteet lähettävät ja keräävät tietoa palvelimelta jopa useita kertoja sekunnissa. HTTP-protokollalla ei voida lähettää tietoa yhdestä pisteestä monelle eri laitteelle, joka on taas IoT-verkoston laajuuden vuoksia hankala toteuttaa. Tapahtumia ei voida seurata silloin kun ne tapahtuvat, jolloin viestinnän varmuus ei ole taattu. Pienien pakettien jakamainen suurissamäärin ja turvallisuus ovat riskitekijä, jos tietoa joudutaan lähettämään turvattomaan verkkoon. Suurimpia heikkouksia on datapakettien koko verrattuna muihin protokolliin, joten IoT-laitteiden virrankulutus ja datan käyttö ovat huomattavasti suurempi tästä johtuen. Protokollan käyttö ei sovi reaaliaikaisen datan lähetykseen. (Karasiewicz 2013.)

4.1.5 CoAP

CoAP (Constrained Application Protocol) protokolla on kehitetty M2M-sovelluksiin, kuten älykkääseen verkkoon ja taloautomaatioon. CoAP-protokolla käyttää sisäänrakennettuja URI-komentoja ja internetin tiedostotyyppejä. CoAP -protokolla on suunniteltu käytettäväksi integroitumalla HTTP-protokollaan, mutta erikoisvaatimuksena monikanavainen keskustelu suljetussa ympäristössä. (CoAP n.d.)

Protokollaa voidaan hyödyntää yksinkertaisille laitteille, kuten valokytkimille, venttiileille ja muille ei niin vaativaa ohjausta käyttäville laitteille, joita voidaan seurata internetissä. CoAP muistuttaa paljon HTTP-protokollaa ja toimii myös samalla asiakas-palvelin mallilla. Eli asiakas joka lähettää pyynnön CoAP-palvelimelle ja josta asiakas saa CoAP-vastauksen. HTTP-protokollasta poiketen, kun viesti on asynkroninen. (Collin & Saarelainen 2016, 188.)

4.2 **Tietokannat**

Tietokanta käsitteellä tarkoitetaan tietokokoelmaa, johon pystytään helposti pääsemään käsiksi, muokkaamaan ja päivittämään. Tietokantoja käytetään tietojen varastointiin, jolloin tietoa voidaan hyödyntää jatkossa. Tietojenvarastointia varten on olemassa erilaisia tietovarastotyyppejä, joista yleisimmät ovat SQL ja NoSQL tietokantoja. Kuinka tietokannat eroavat toisistaan ja kuinka tietokantaa tietyntyyppistä tietokantaa voidaan hyödyntää IoT-laitteiden datankeruussa.

4.2.1 SQL

SQL (Structured Query Language) on tietokannoissa käytettävä standardoitu kieli, jonka ensimmäinen versio syntyi vuonna 1974, mutta tuolloin kieltä kutsuttiin SEQUEL-kieleksi, jonka nimi myöhemmin muuttui SQL:ksi. Ensimmäinen virallinen SQL-standardi julkaistiin vuonna 1986. (Lehtonen 2002, 38)

Lehtosen (2002, 38) mukaan SQL-muotoa ei ennen SQL99 version tulemista standardoitu vaan silloin tietokannat olivat pitkään valmistajien omina versioina. SQL kieli sisältää seuraavat ominaisuudet:

- tietokannan rakenteen määrittelyn ja muuttamisen
- kyselyjen tekemisen
- tietojen lisäämisen, muuttamisen ja poistamisen
- tapahtumakäsittelyn ohjaamisen
- upotetun SQL:n ja kohdistimien hallitsemisen

Maailman suurimpia tietokantoja on jo pitkään ollut MySQL, Oracle ja Microsoft SQL Server. Nämä tietokannat noudattavat rakenteellista SQL-mallia. Tämä edellyttää, että data viedään tietovarastoon ennalta määrätyssä muodossa, eli data tallennetaan taulukkoon. Tämän tyylinen tietokanta ei ole IoT-laitteiden laajennettavuuksien vuoksi kovin hyvä vaihtoehto tietokannaksi. Kun datavirrat kasvavat perinteinen SQL-tietokanta hidastuu ja tietokanta voi muuttua käyttökelvottomaksi. (Collin & Saarelainen 2016, 197.)

SQL-tietokannan käyttö voi monessa sovelluksessa olla sopiva vaihtoehto, jos data ei ole kovin strukturoitua ja taulut ovat oikein skaalattu ja indeksoituna. Mutta jos myöhemmin on tarvetta skaalata järjestelmää, niin silloin SQL- tietokanta ei ole hyvä vaihtoehto. Silloin vastaan tulevat ongelmat määrättyjen tiedostomuotojen kanssa, joka vaikeuttaa datan lukua ja tallentamista tietokantaan. (Collin & Saarelainen 2016, 199.)

4.2.2 NoSQL

NoSQL-tietokanta on asynkroninen tietokanta, joka on todella skaalautuva ja joustava. NoSQL-tietokanta sallii tiedostojen tallentamiseen ilman pilkkomista tai datan muuttamista erimuotoiseksi. Joten NoSQL-tietokanta sallii tallentamaan samaan tietokantaan strukturoitua ja strukturoimatonta dataa. Tämä auttaa tietokannan ja ohjelmistokehityksessä paljon, joten ei enää tarvitse datan rakennetta päättää etukäteen. (Collin & Saarelainen 2016, 197)

Collin & Saarelainen (2016, 198.) mukaan NoSQL-tietokantoja on olemassa markkinoilla yli sata erilaista, joiden ominaisuudet poikkeavat toisistaan. Ne voidaan luokitella neljään eri ryhmään:

- dokumenttirakenteiset
- laajasarakkeiset
- avainarvoperusteiset
- graafiset

Maailman suosituin NoSQL-tietokanta on dokumenttirakenteinen MongoDB, jonka ominaisuuksiin kuuluvat dokumenttien tallentaminen JSON-rakennemuotoon, jota käytetään yleisesti ohjelmoinnissa ja rajapinnoissa. Tämä helpottaa sovelluskehittäjän tietokannasta luettavan datan muotoilua. (Collin & Saarelainen 2016, 198.)

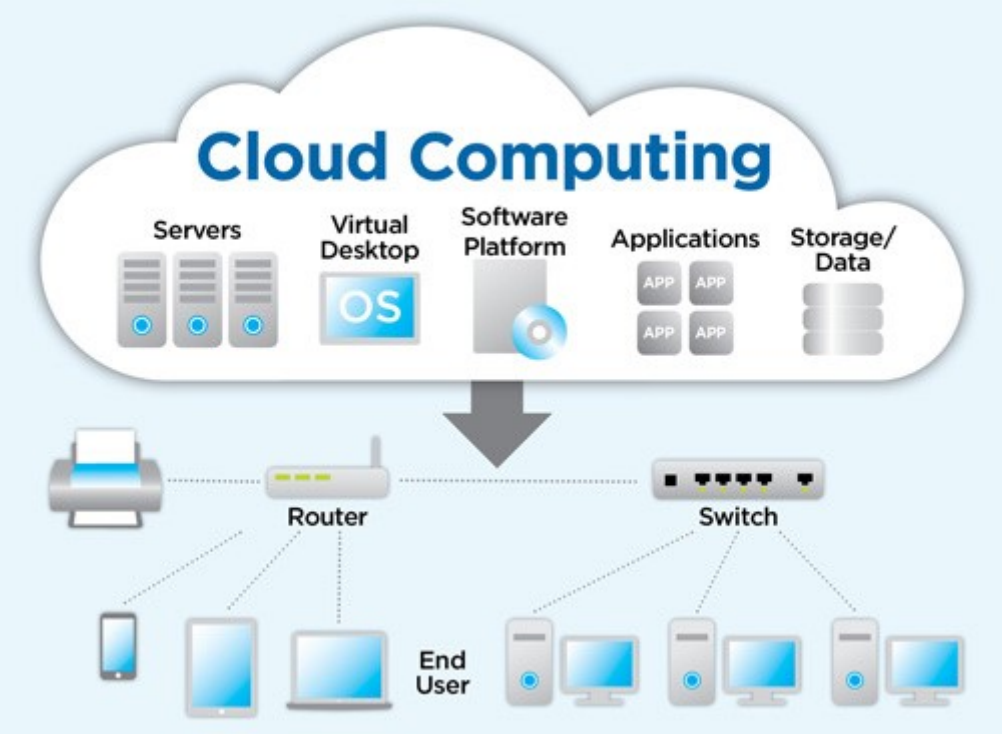
NoSQL-tietokanta sopii IoT-laitteiden tiedon tallentamiseen skaalautuvuuden vuoksi, koska ennalta määrättyä muotoa ei ole, vaan IoT-laitteelta syötettävä data voidaan suoraan syöttää NoSQL-tietokantaan, jolloin vältytään turhalta tiedostojen muokkaamiselta.

5 PILVIPALVELUN TARJOAJAT

5.1 Pilvipalvelun määrittely

Pilvipalvelulla tarkoitetaan palveluntarjoajan verkkopohjaista alustaa, jossa on erilaisia sovelluksia erilaisiin tarkoituksiin. Tärkeimmät palvelut ovat tietokannat ja niille tarkoitetut erilaiset tiedonkäsittelyyn tarvittavat sovellukset. Pilvipalvelu on saanut nimensä siitä, että arkkitehtuurikuvissa

yleensä internetiä kuvataan pilvisymbolilla. Kuten Kuvassa 5 voidaan huomata, että pilvipalvelussa on erilaisia palveluita, joita loppukäyttäjä voi hyödyntää. (Harvey, C 2017.)



Kuva 5. Pilvipalvelun esimerkkikuva (Harvey, C 2017.)

Esineiden internetiä varten myös monella palveluntarjoajilla on käytössä IoT-alusta, johon voidaan erilaisia laitteita ja antureita liittää turvallisesti. Monilla palvelun tarjoajilla on käytössään uusimmat viestintäprotokollat ja viimeisimmät turvallisuutta parantavia tekijöitä.

Palveluntarjoajilla on yleensä käytössä endpoint nimellä kutsuttuja asetuksia, jolla tuleva data voidaan ohjata haluamaansa pilvipalvelun sovellukseen. Palveluntarjoajilla on käytössä datan varastointiin käytettävät tietokannat, jotka ovat soveliaita suurien datamassojen tallennukseen ja sovelluksia datan analysointia varten. Seuraavaksi käydään läpi työhön sopivia pilvipalveluntarjoajia, joilta löytyy IoT-sovellukseen sopivia ratkaisuja ja sovelluksia.

5.2 **Amazon web services**

Amazon web services (AWS) on vuonna 2006 aloitettu palvelu, joka tarjoaa verkkopohjaisia infrastruktuurisia palveluita yrityksille. Tämän ansiosta yrityksien ei enää tarvitse suunnitella ja hankkia omia palvelimia. (AWS n.d.).

Amazonin pilvipalveluun kuuluvat kaikki IoT-sovelluksen luomiseen tarvittavat työkalut. Amazonin palveluihin kuuluvat AWS IoT Core -palvelu, johon voidaan liittää laitteita lähettämään tietoa. Dataa lähetettäessä IoT-

palveluun voidaan datan kulkua reitittää sisään rakennetuilla käskyillä menemään tietokantaan, josta siitä voidaan myöhemmin käyttää analysoimiseen ja tiedon näyttämiseen erilaisissa muodoissa eri sovelluksien avulla.

Amazonin pilvipalvelua voi testata vuoden ajan ilman kustannuksia. Tietenkin, jos haluaa käyttää ominaisuuksia, jotka eivät kuulu ilmaisen kokeilun piiriin, joutuu siitä maksamaan erikseen määritellyn summan. Ilmaiseen testaukseen kuuluu kaikki palvelut, mutta ilmainen datakeruu loppuu 25 GB jälkeen. Kun tämä määrä ylitetään, joudutaan ylityksestä maksamaan tietty summa. (AWS n.d.)

Amazonin IoT-palvelun hinnoittelu perustuu laitteiden määrään, laitteiden lähettämien viestien määrään ja laitteiden hakemiin tietoihin IoT-palvelusta. Asiakkaan laskutus tapahtuu kuukausittain. (AWS n.d.)

AWS IoT Core tukee MQTT- ja HTTP-protokollaa, joilla laitteet voidaan liittää palveluun. Palvelu tarjoaa turvallisen liikennöinnin laitteen ja pilvipalvelun välillä. Laitteita voidaan liittää palveluun ilman rajoitteita. (AWS n.d.)

5.3 **Mizrosoft Azure**

Microsoftin kehittämä Azure IoT Hub on pilvipohjainen alusta, joka mahdollistaa laajan hallittavuuden ja tietoturvallisen ympäristön miljoonien eri IoT-laitteiden ja eri tietoratkaisujen välillä. Microsoft tarjoaaa muun muassa luettavan laitteiden ja pilvipalvelun välisen viestinnän, turvallisen kommunikoinnin laitteella ja laajan tuen erityyppisille laitteille. (Microsoft 2016.)

Microsoftin Azure on pilvipalvelun ydin, jonka kautta pystytään luomaan ja ottamaan käyttöön eri palveluita, joita voidaan tarvita esimerkiksi datan varastointiin tai datan analysointiin. Microsoft Azuren kautta voidaan luoda virtuaalikoneita, SQL-tietokantoja ja NoSQL-tietokantoja. Azuren hinnoittelu perustuu ”pay-as-you-go” tapaan, jossa käyttäjää laskutetaan käytön mukaan tunneittain.

IoT-hub hinnoittelu riippuu käytettävästä versiosta, joita Microsoft tarjoaa 4 erilaista, jotka ovat kuvassa 4. Tarjontaan kuuluu ilmainen versio, joka mahdollistaa sovelluksien testaamista pienessä mittakaavassa.

EDITION TYPE	PRICE PER UNIT (PER MONTH)	TOTAL NUMBER OF MESSAGES/DAY PER UNIT	MESSAGE METER SIZE
Free	Free	8,000	0.5 KB
S1	€42.17	400,000	4 KB
S2	€421.65	6,000,000	4 KB
S3	€4,216.50	300,000,000	4 KB

Kuva 6. IoT-Hub hinnoittelu (IoT Hub n.d.)

IoT Hub tukee monien eri laitteiden yhdistämistä jo valmiiksi, etenkin MQTT- ja AMQP-protokollaa käyttävien laitteiden liittämistä, mutta Microsoft tarjoaa kuitenkin työkaluja eri protokollaa käyttävien laitteiden yhdistämiseen.

Microsoft Azuren portaalin käyttäminen toimii siten, että valikosta valitaan haluttu sovellus. Valittu sovellus aukeaa samaan ikkunaan, mutta uudelle lehdelle. Tämä ominaisuus on toimiva, mutta virhepainalluksen sattuessa asetuksien laatimisen voi joutua aloittamaan uudestaan. Microsoftin tarjoaa laajoja ohjedokumentaatioita, joiden avulla sovelluksien käyttöönotto onnistuu pienen opiskelun jälkeen.

5.4 **Google CloudPlatform**

Googlen kehittämä CloudPlatform, joka on Microsoftin ja Amazonin rinnalla kaikista uusin tulokas IoT -maailmaan. Google Could IoT core mahdollistaa datan monitoroinnin ja analysoinnin reaaliajassa. (Google a, 2017.)

Google lupaa nopean, helpon ja hallittavan laitteiden liitettävyyden. Laitteiden liittäminen palveluun tapahtuu joko MQTT- tai HTTP-protokollalla. IoT Core mahdollistaa tietojen lähettämisen eteenpäin muihin Google CloudPlatform -palveluihin. (Google a, 2017.)

IoT Coren hinnoittelu perustuu kuukausittaiseen data määrään, joka kertyy lähetettävistä viesteistä. Ensimmäiset 250 megatavua on ilmaista viestintää, jonka ylityttyä jokaiselle megatavulle joutuu maksamaan tietyn summan riippuen kuukaudelle varatusta määrästä. Google käyttää ”pay-as-you-go” tapaa, jolloin käyttäjä maksaa palveluiden käytöstä tunneittain. (Google b, 2017.)

6 **PILVIPALVELUN JA PROTOKOLLAN VALINTA**

Pilvipalvelun valinta aloitettiin käymällä läpi pilvipalveluntarjoajia ja niiden tarjoamia sovelluksia ja hinnoittelua ja palvelun tarjoajan tukemia protokollia, joista valittiin sopivin vaihtoehto tähän työhön. Tässä kappaleessa käydään läpi eri palveluntarjoajien mahdollisuuksia ja perustelut, miksi päädyttiin käyttämään tietyn tarjoajan pilvipalvelua.

Protokollan kohdalla valintakriteerinä oli mahdollisimman kevyt, turvallinen ja laajennettava vaihtoehto, jolloin sovellus olisi helposti toteutettavissa. Viestinnän tulisi olla mahdollisimman tehokasta ja sopia mahdollisimman monelle eri laitteelle.

6.1 **Pilvipalvelu**

Pilvipalvelun lopulliseen valintaa vaikuttavia tekijöitä olivat laajennettavuus, monipuoliset liitännät IoT-ympäristöön ja kustannukset. Tässä vertailussa Microsoftin Azuren pilvipalvelu osoittautui parhaaksi vaihtoehdoksi. Sillä on monipuoliset viestintäprotokollien liitännäisyydet ja monipuoliset sovellukset erilaisten toimenpiteitten suorittamiseksi. Palvelusta löytyy viestinnälle tärkeät ominaisuudet, kuten viestintä laitteiden välillä ja laitteelta tietokantaan, joten tämä on sopiva sovelluksen toteuttamiseen.

6.2 **Protokolla**

Taulukossa 1 on käyty läpi protokollien eroavaisuuksia, jonka pohjalta tehdään valinta, mikä viestintäprotokolla sopii sovelluksen toteuttamiseksi.

Taulukko 1. Viestintäprotokollien eroja (IoT application protocols 2016.)

Protokolla	MQTT	CoAP	XMPP	AMQP
Viestintämalli	Julkaisija/tilaaja	Pyyntö/vastaus	Pyyntö/vastaus tai Julkaisija/tilaaja	Julkaisija/tilaaja
Turvallisuus	Valinnainen	Valinnainen	Korkea	Korkea
Viestinkoko	2 tavua	4 tavua	-	8 tavua
Viestintävarmuus	Korkea	Keskikertainen	-	Korkea
Virrankulutus	Pieni	Keskinkertainen	Korkea	Keskinkertainen

Jokaisella viestintäprotokolalla on omat käyttötarkoituksensa. Tässä työssä taulukon 1 perusteella päädyttiin käyttämään MQTT-protokollaa, koska sen avulla sovellus on mahdollisimman helposti toteutettavissa ja jatkossa erilaisten laitteiden lisääminen ei tuota ongelmia protokollan helppouden takia. Myös suuren suosion ansiosta, suurin osa ohjelmointikielistä ja teollisuudessa käytettävistä logiikoista hyödyntävät MQTT-protokollaa sen keveyden ja viestintä varmuuden vuoksi.

7 **IOT SOVELLUS**

Sovelluksessa oli käytössä Microsoft Azure pilvipalvelu, jossa on tarvittavat sovellukset IoT-järjestelmän toteutukseen. Tietojen välitykseen laitteelta

pilvipalvelimelle tarvitaan käyttöön sovellus, jolla tiedot lähetetään IoT-alustalle, joka on palvelussa. Esimerkkisovelluksessa on käytössä Node-RED kehitysympäristö, joka on asennettuna Raspberry Pi-tietokoneelle, jotka käydään läpi myöhemmin.

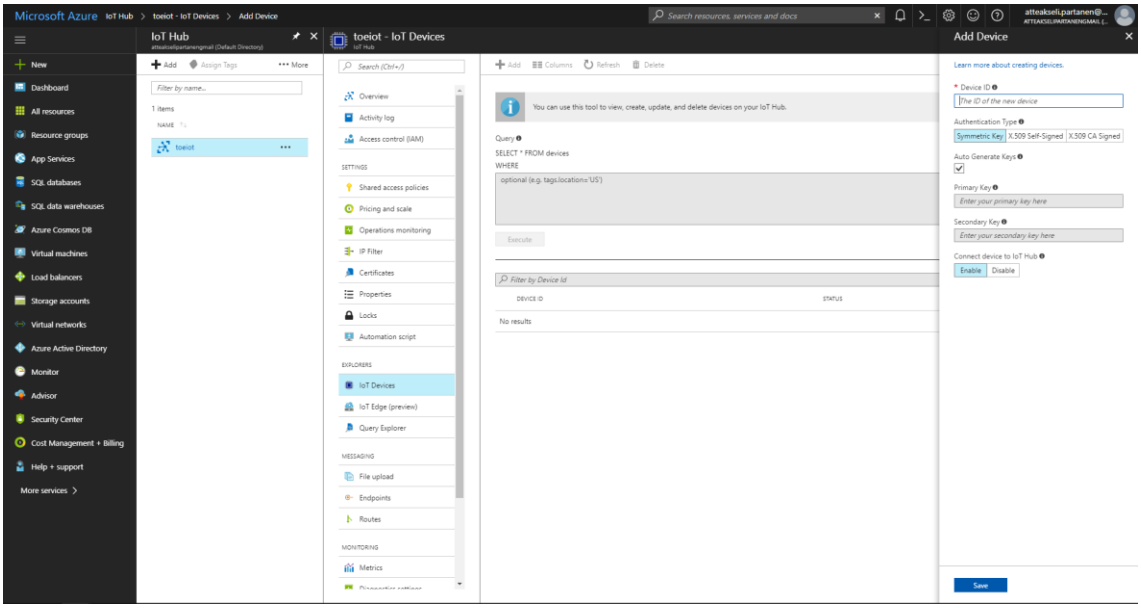
Esimerkkisovelluksessa on käytössä Microsoft Azure pilvipalvelu, jonka palvelut mahdollistavat anturin lähettämän tiedon lukemisen ja tiedon tallentamisen tietokantaan ja sieltä lukemiseen. Seuraavaksi käydään läpi käytettyjen palveluiden ominaisuuksia ja niiden tarpeellisuutta sovelluksessa. Työssä on käytössä seuraavat Microsoft Azuren tarjoamat pilvipalvelut:

- IoT Hub, laitteen liittäminen pilvipalveluun
- Streaming Analytics, laitteen datan siirtäminen haluttuun palveluun
- DocumentDB, NoSQL tietokanta, johon tiedot tallennetaan

7.1 IoT Hub

Microsoft Azuren tarjoamaan ilmaiseen IoT Hub versioon voidaan liittää 500 laitetta ja laitteilla voidaan lähettää yhteensä 8000 viestiä, jotka riittävät testausympäristön valmistamiseen mainiosti. Ilmaisessa suunnitelmassa on myös mukana yksi kustomoitava päätepiste, joka avulla voidaan laitteiden mittaama data siirtää muihin palveluihin.

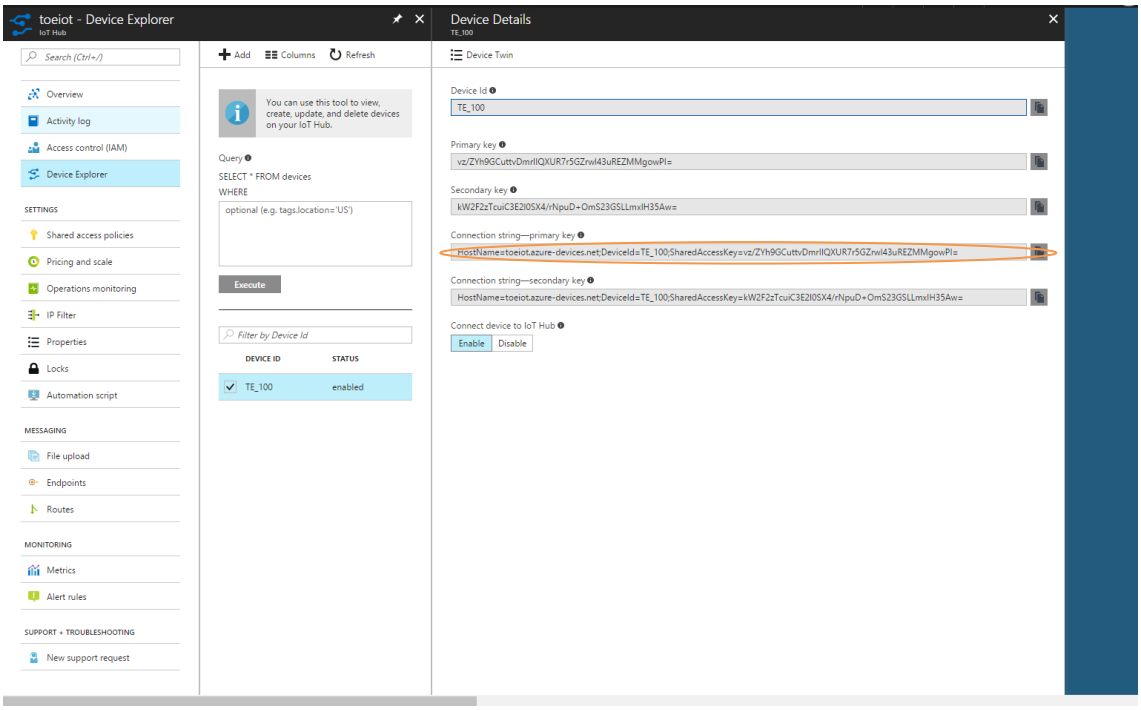
Uuden laitteen lisäys IoT Hubiin voidaan hoitaa verkkosivun kautta kuvassa 7, jolloin laitteelle muodostetaan yhdistämisavaimet, turvallista liikennöintiä varten.



Kuva 7. IoT Hub uuden laitteen lisäys

IoT Hubiin on luotu TE_100 identiteetillä oleva laite, jolle voidaan lähettää tietoa yhteydenmuodostus osoitteella, joka on ympyröitynä kuvassa 8.

Osoite koostuu isäntäosoitteesta, laite identiteetistä ja yhdistämisavaimesta.



Kuva 8. IoT Hub laitteen tiedot

IoT Hubin kautta laitteelle voidaan määrittää erilaisia endpoint nimellä kutsuttuja pisteitä, mihin laitteilta kerätty data jatketaan, koska pelkkä IoT Hub itsessään ei kerää tietoa, vaan se välittää tiedon halutulle palvelulle. Vakiona IoT Hub antaa jokaiselle laitteelle kaksi valmiiksi asetettua endpointin, jotka ovat pilvestä laitteelle palaute ja myös viestintä Event Hub-palveluun, josta voidaan muodostaa erilaisia hälytyksiä.

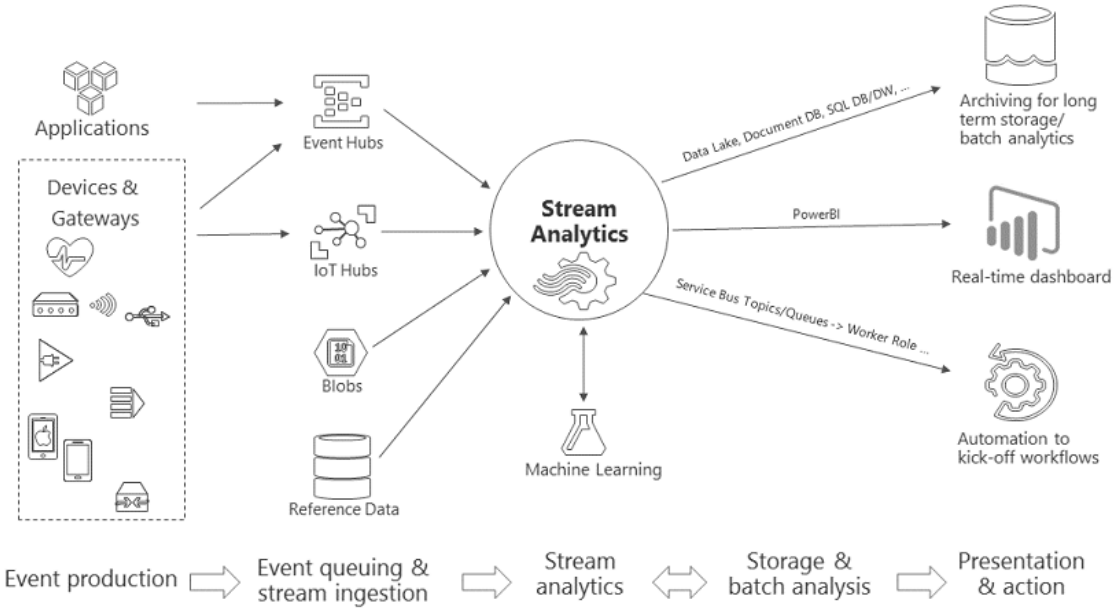
Liikennöinti laitteen ja palvelimen välillä tapahtuu luodun yhdistysavaimen ja IoT Hub-palvelin osoitteen avulla, jotka tarvitaan laitteen yhdistämistä varten ohjelmistoa tehdessä. IoT Hubiin voidaan siirtää tietoa MQTT-, AMQP- ja HTTP-protokollilla, jotka ovat kaikista yleisimmät viestintä protokollista.

Jotta IoT Hub:iin tulevien antureiden tiedot saadaan siirrettyä eteenpäin, tarvitaan Microsoft Azuren palveluista käyttää Streaming Analytics-palvelua, jonka avulla voidaan mitatut tiedot välittää haluttuun paikkaan SQL-queryn avulla.

7.2 **Streaming Analytics**

Microsoft Azuren Streaming Analytics -palvelu on suunniteltu siirtämään suuria määriä tietoa haluttuun paikkaan. Tietojen siirto tapahtuu lähes reaaliajassa, joten tämä takia sitä käytetään monessa tapauksessa tietojen välittämiseen haluttuun palveluun, kuten kuvassa 9 on näytetty. Tämän

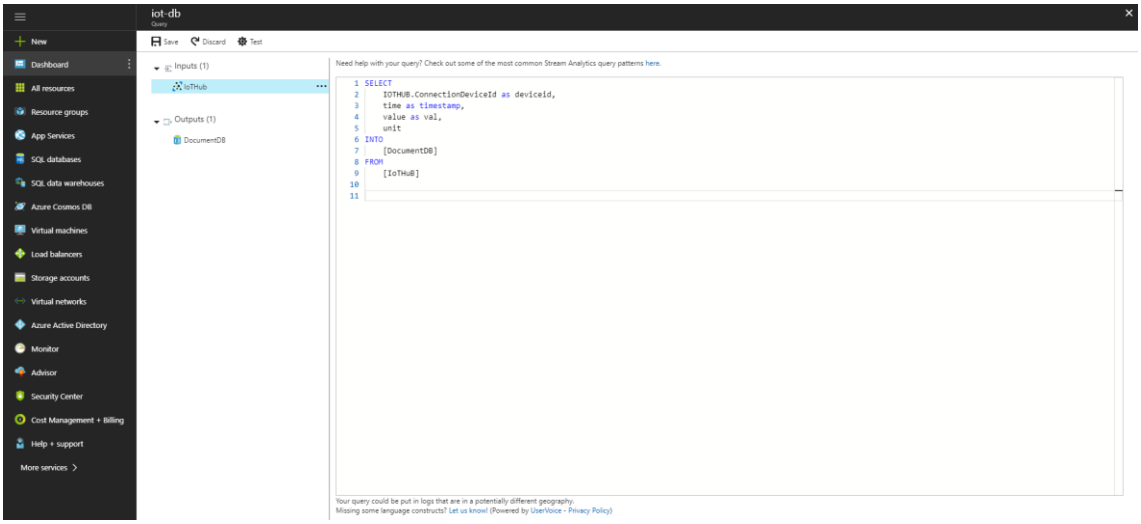
ansiosta voidaan tulevaisuudessa tarpeen vaatiessa lisätä palveluita, joihin data voidaan siirtää tai vaihtaa.



Kuva 9. Stream Analytics sovelluksia (Microsoft 2017.)

Stream Analytics käyttö monessa IoT-sovelluksessa on tarpeen, koska laitteelta saatava mittaustieto halutaan tallentaa tietokantaan, josta sitä voidaan myöhemmin analysoida. Stream Analytics avulla voidaan data siirtää palveluihin, jossa dataa tarvitaan ja voidaan hyödyntää ohjauksissa. Stream analytics toimii välittäjänä, jonka avulla tiedon kulkua voidaan ohjata haluttuihin palveluihin.

Stream Analytics tiedonvälityksen konfigurointi hoidetaan SQL-querylla. Queryn avulla voidaan haluttu tieto datasta siirtää haluttuun paikkaan. Jotta halutut tiedot saadaan lähetettyä haluttuun paikkaan, tarvitaan Stream Analytics palvellun quary määrittää kuvan 10 mukaisesti. Siinä on määrätty, että tulevan data on peräisin IoT Hubsta. Kun tuleva tieto luetaan, voidaan siitä valita halutut tiedot, jotka välitetään ulostuloon, johon on määritelty tietokanta.



Kuva 10. Stream Analytics query

Ilman minkäänlaista datan käsittelyä ennen tietokantaan tallennusta tallennettava data näyttää kuvan 11 mukaiselta. Turhat tiedot tallentuvat myös tietokantaan, joka lisää tietokannasta lukiessa lukemisen määrää ja tallennettavan dokumentin kokoa, joka kasvattaa sen lukemisen ja tallentamisen kustannuksia.

TIME	VALUE	UNIT	EVENTPROCESSEDUTCTIME	PARTITIONID	EVENTENQUEUEDUTCTIME	IOTHUB
1505305744372	26.02	"celsius"	"2017-09-13T12:29:30.0094508...	1	"2017-09-13T12:29:03.4970000...	{"MessageId":null,"CorrelationI...
1505305745374	27.68	"celsius"	"2017-09-13T12:29:30.0094508...	1	"2017-09-13T12:29:04.4910000...	{"MessageId":null,"CorrelationI...
1505305746375	29.74	"celsius"	"2017-09-13T12:29:30.0094508...	1	"2017-09-13T12:29:05.5110000...	{"MessageId":null,"CorrelationI...

Kuva 11. Käsittelemätön data

Kun data käsiteltiin kuvan 10 mukaisella querylla saatiin datasta turhat tiedot suodatettua pois, jolloin säästyttiin turhien tietojen tallentamiselta tietokantaan. Kun data käsitellään kuvan 12 mukaiseen muotoon, voitiin data välittää toiseen palveluun.

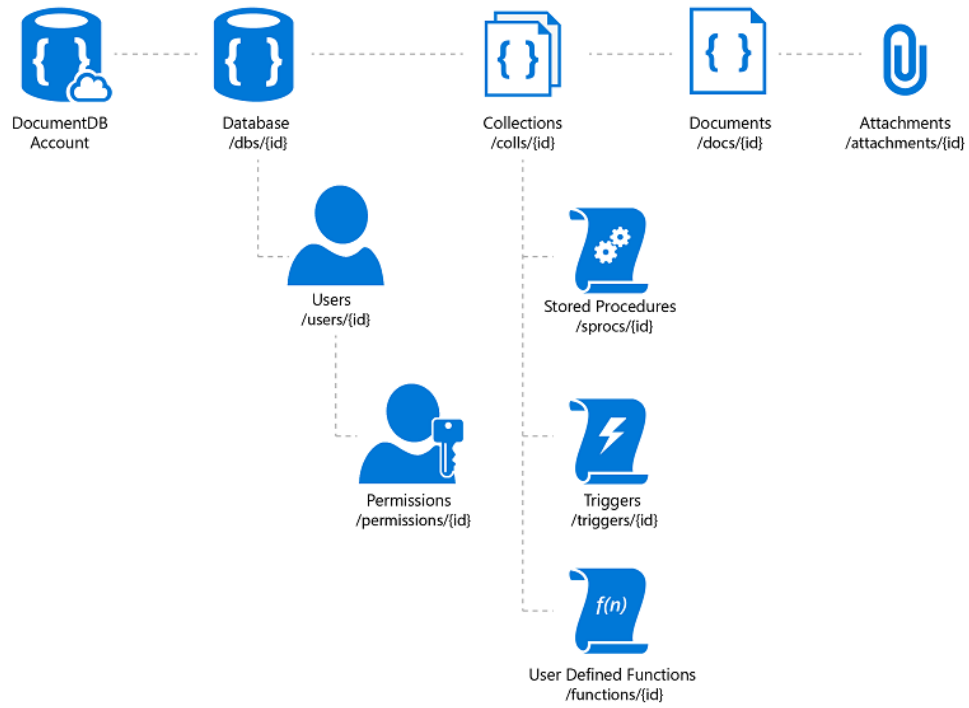
DEVICEID	TIMESTAMP	VAL	UNIT
"TE_100"	1505305744372	26.02	"celsius"
"TE_100"	1505305745374	27.68	"celsius"
"TE_100"	1505305746375	29.74	"celsius"
"TE_100"	1505305747383	29.7	"celsius"
"TE_100"	1505305748385	29.08	"celsius"

Kuva 12. Käsitelty data

7.3 DocumentDB

Microsoft Azuren tarjoama DocumentDB ohjelmarajapinta on NoSLQ-tietokanta. DocumentDB tietokanta on dokumenttipohjainen ja kaikki dokumentit tallennetaan JSON-muodossa, joka helpottaa datan käsittelyä jatkossa. Dokumentti sisältää tietokannan nimen ja sille kohdistetun kokoelma nimen ja identiteetin, johon tiedosto halutaan tallettaa, lukea, päivittää tai poistaa. Tietokannan lukemiseen käytetään SQL-querya, jolla voidaan halutut tiedot lukea tietokannasta.

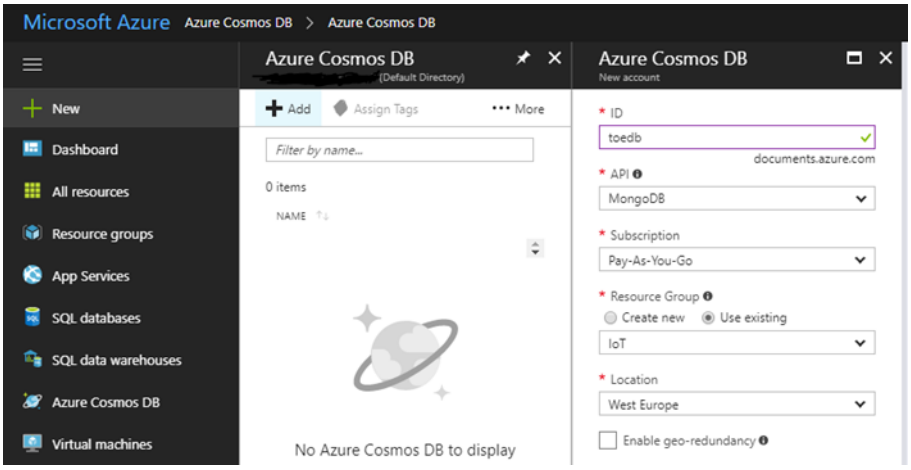
DocumentDB tietokannan rakenne on monikerroksinen, joka on esitelty kuvassa 13. Tietokannassa voi olla monta eri käyttäjää, joilla on tietyt pääsy oikeudet tiettyyn kokoelmaan.



Kuva 13. DocumentDB rakenne (Microsoft DocumentDB 2017.)

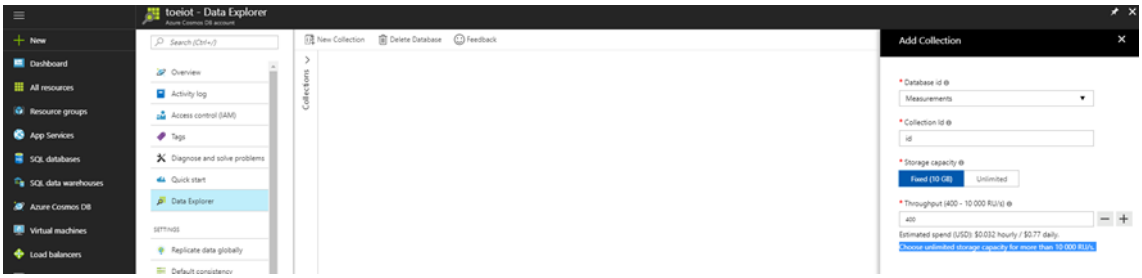
Tietovarastoa luodessa valitaan tietovaraston koko ja kuinka paljon tiedostoja pystytään tallentamaan ja lukemaan tietovarastosta. RUn (request units) avulla määritetään, kuinka paljon tietoa voidaan maksimissaan lukea tietokannasta ja kuinka paljon tiedostojen lukemisesta maksetaan.

Tietokanta voidaan luoda Microsoft Azuren portaalin kautta kuvassa 14, jossa määritellään tietokannan identiteetti, minkälaisessa muodossa tietokanta on ja muut tarvittavat asetukset, kuten minne alueelle tietokanta halutaan luoda ja minkälaisilla oikeuksilla.



Kuva 14. Tietokannan luonti

Kun tietokanta on luotu, voidaan portaalin kautta lisätä kokoelma kuten kuvassa 15, johon kerätään tietoa identiteetin avulla. Kokoelman luonti sivulla näytetään, kuinka paljon kokoelman ylläpitäminen maksaa tunnissa.

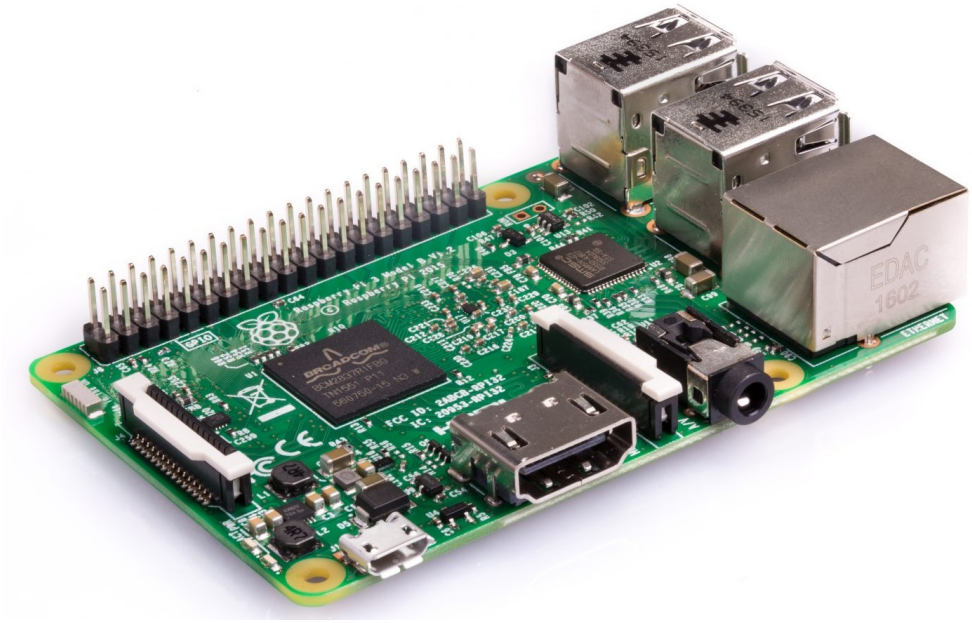


Kuva 15. Kokoelman luominen

7.4 **Raspberry Pi**

Raspberry Pi on avoimeen lähdekoodiin perustuva yhden piirin tietokone, jonka on kehittänyt brittiläinen Raspberry Pi Foundation. Tarkoituksena on tuoda halpa ja tehokas tietokone, jota voidaan hyödyntää sovelluksissa ja opetustarkoituksissa. (Raspberry n.d.)

Raspberry Pi tietokoneesta on tullut kolme eri sukupolvea, joista uusin on Raspberry Pi 3 Model B, joka on kuvassa 16. Edellisiin sukupolviin verrattuna uusin malli on kaikista nopein ja tietokoneessa olevien liitäntöjen määrä on kasvanut paljon. Raspberry Pi 3-malliin on sisäänrakennettu WLAN-sovitin ja Bluetooth, joiden ansiosta ei enää tarvitse käyttää rajoitettuja USB-portteja yhteyksien lisäämiseksi. (Raspberry n.d)



Kuva 16. Raspberry Pi 3 Model B (Raspberry n.d.)

Raspberry Pi käyttöjärjestelmänä toimii monia eri linux-pohjaisia vaihtoehtoja, joista pääsääntöisesti käytetään Raspberry Pi Foundation räätälöimää Rasbian-käyttöjärjestelmää, joka on kevyempi versio Debian-käyttöjärjestelmästä. Käyttöjärjestelmän asentaminen tapahtuu microSD-muistikortille, jonka minimi koko on 8GB, kun itse asennuspaketin koko on minimissään 4GB. (SD card n.d.)

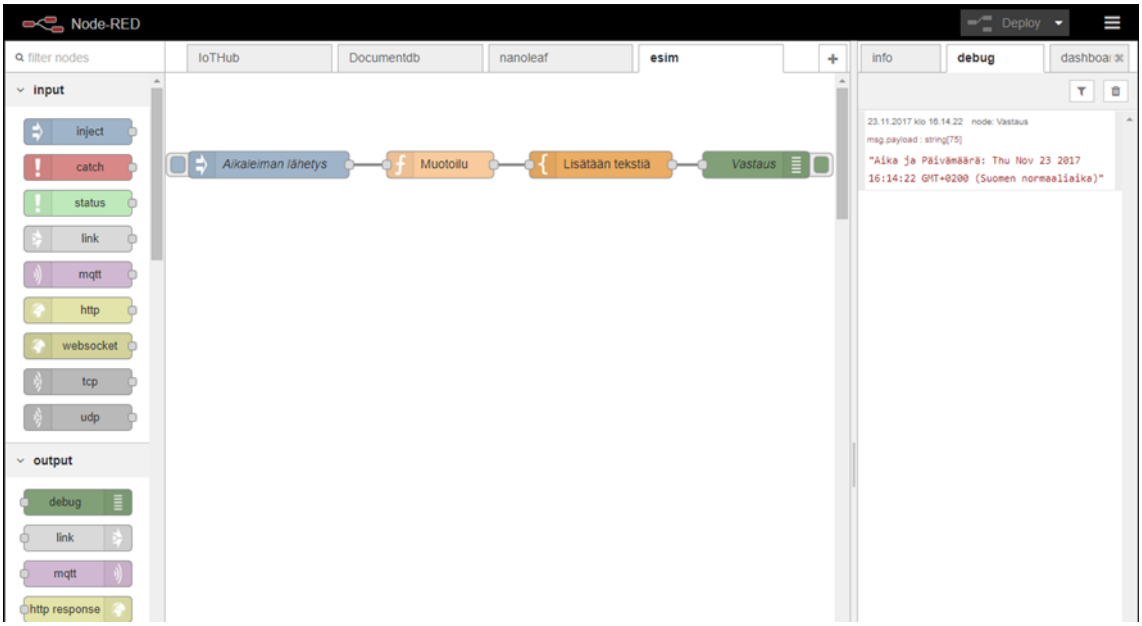
7.5 **Node-RED**

Node-RED käyttö tässä työssä perustuu sen helppouteen ja käytettävyyteen esineiden internetin ohjelmoinnissa. Node-RED koostuu kirjastopaketeista, jotka on kehitetty tiettyä toimintoa varten. Node-RED sisältää sovellukseen tarvittavat liikennöintikirjastot esineiden internetin muodostamista varten. Node-RED on mahdollista asentaa monille eri käyttöalustoille kuten Raspberry pi, Android ja Arduino. Monet pilvipalvelun tarjoajien virtuaalikoneille voi asentaa oman Node-RED kehitysympäristön. (Node-RED n.d.)

Node-RED on J. Paul Morrisonin kaaviomuotoisen ohjelmoinnin pohjalta tehty ohjelmointikieli, jossa sovelluksen toimintoja kuvataan laatikoina. Jokaiselle laatikolle on annettu tärkeä tehtävä, jonka se suorittaa ja välittää eteenpäin. Laatikoiden verkosto vastaa tietojen kulusta. (Node-RED n.d.)

Node-RED koostuu Node.js-pohjaisesta runtime-ohjelmasta, johon pääsee käsiksi verkkoselaimen kautta. Selaimen kautta voi luoda sovelluksen vetämällä laatikoita työtilaan ja liittää ne yhteen. Kun haluttu sovellus on tehty, voidaan sovellus suorittaa lähettämällä työtila runtime-ohjelmalle, joka rupeaa suorittamaan sovellusta. (Node-RED n.d.)

Node-RED:ä käytetään asettamalla ohjelmistossa tarjolla olevia laatikoita, joita voidaan linkittää toisiinsa. Kuvassa 17 olevassa esimerkissä on tehty ohjelma, joka tulostaa ajan ja päivämäärän.



Kuva 17. Node-RED esimerkki sovellus

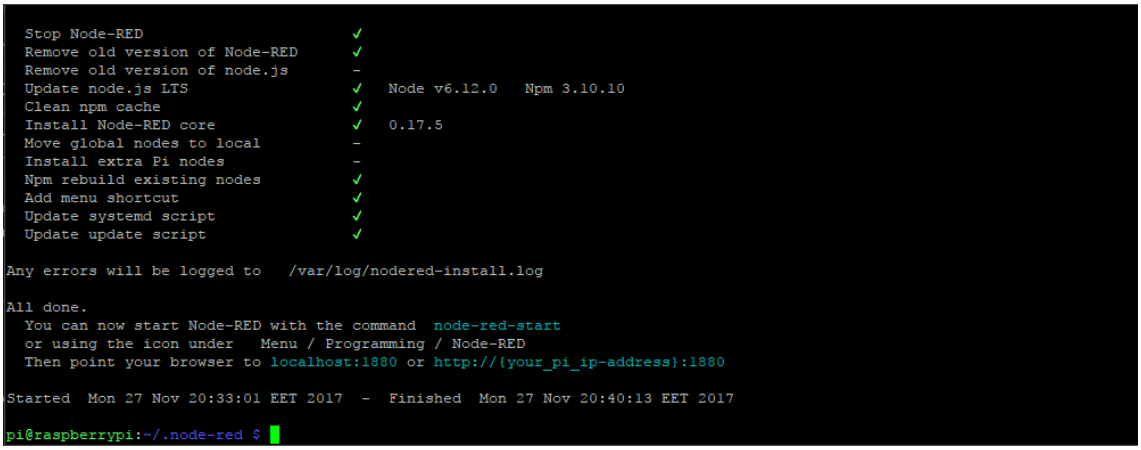
Työtilassa voidaan vasemmalta sivulta olevista laatikoista kuvassa 17 rakentaa haluamansalainen sovellus liittämällä laatikkoja toisiinsa. Node-REDiin on myös mahdollista lisätä yhteisön tekemiä kirjastoja, joita voi hyödyntää omassa ohjelmassa. Yksi suurimpia vahvuuksia on monien eri tiedostoprotokollien käsittely ja monipuolinen liitettävyys erilaisiin tilainteisiin.

8 ESIMERKKISOVELLUS

Sovelluksen toteutus aloitettiin asentamalla Node-RED kehitysympäristö Raspberry Pi-tietokoneelle, jossa käyttöjärjestelmä oli valmiiksi asennettuna. Node-RED asentaminen tapahtui komennolla:

a$ bash <(curl -sLhttps://raw.githubusercontent.com/node-RED/Raspbian-deb-package/master/resources/update-nodejs-and-nodeRED)

Tämä käynnistää asennuksen, joka näyttää kuvan 18 mukaiselta.



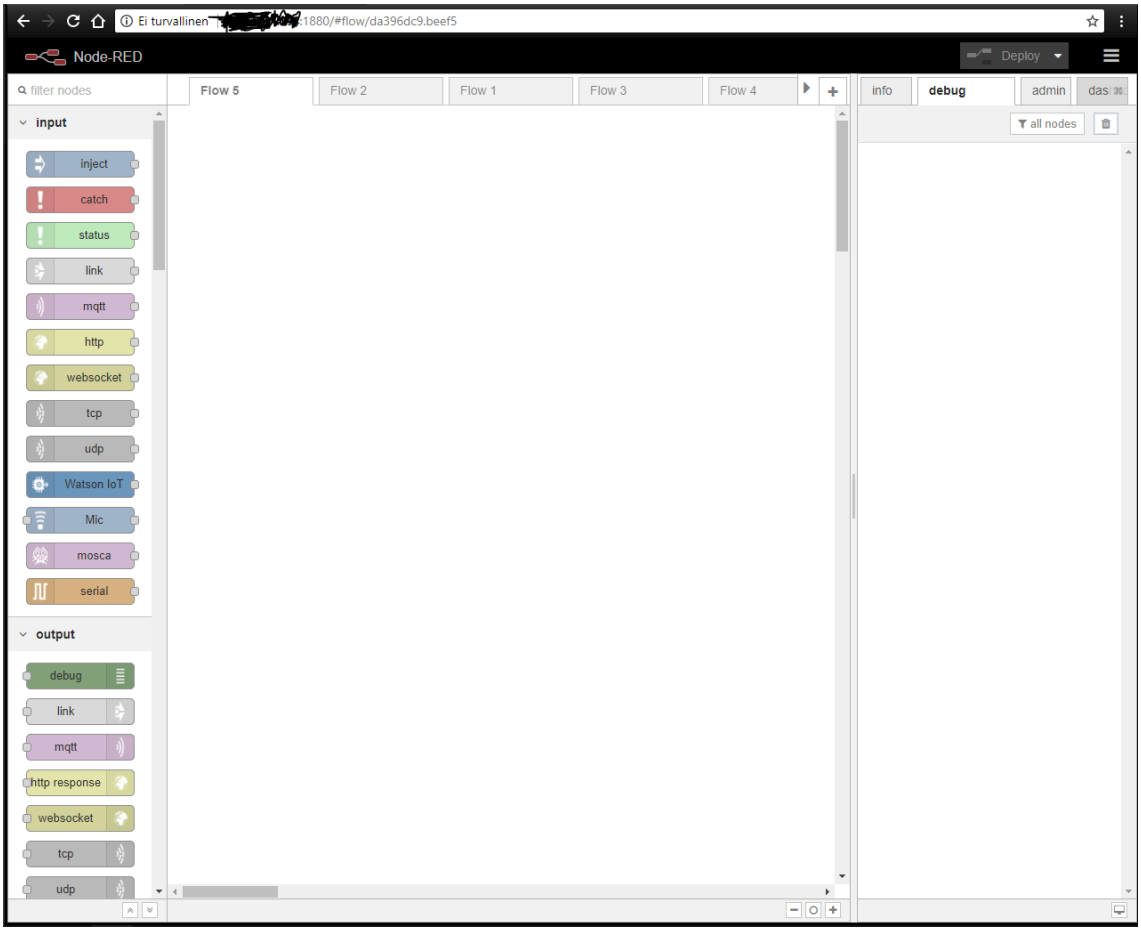
Kuva 18. Node-RED asennus

Kun asennus on suoritettu loppuun, voidaan Node-REDin käynnistäminen varmistamiseksi käyttää komentoa:

$sudo systemctl enable nodeRED.service

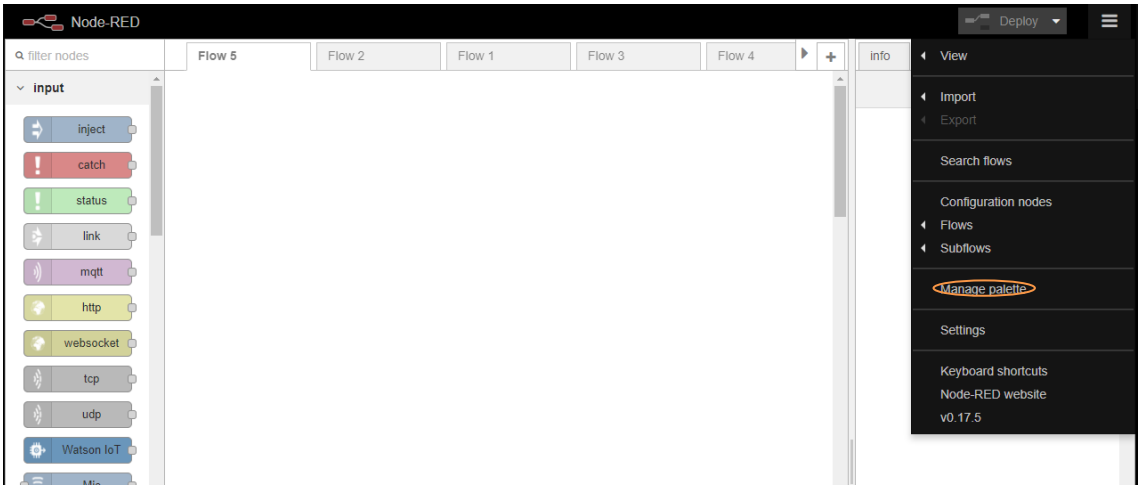
Komennolla Node-RED muuttuu palveluksi, joka suoritetaan aina, kun tietokone käynnistyy udelleen.

Kun Node-RED asentaminen oli suoritettu loppuun, voitiin aloittaa Node-RED kehitysympäristön käyttäminen. Ohjelmointisivu aukeaa laitteen IP-osoitteen ja verkkoportin 1880 avulla, jolloin aukeaa kuvan 19 mukainen verkkosivu.



Kuva 19. Node-RED verkkosivu

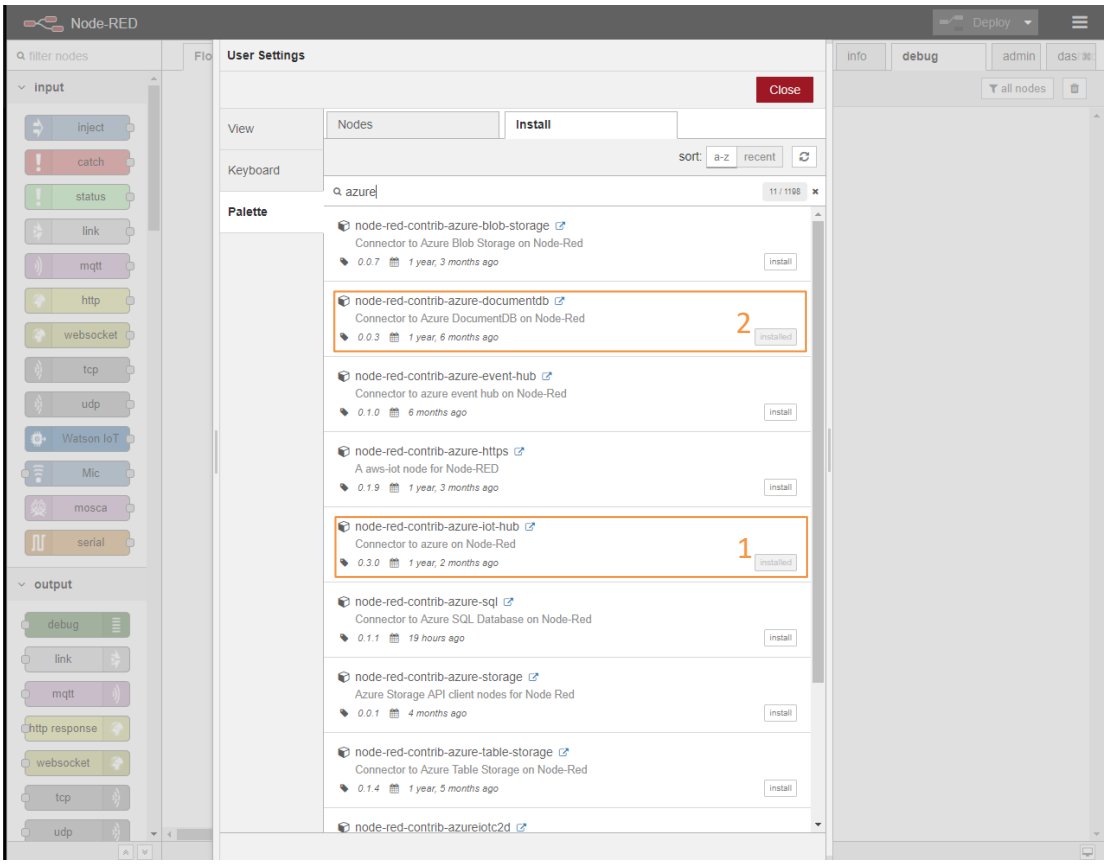
Jotta laitteilta mitatut tiedot saatiin lähetettyä pilvipalvelimelle, tarvittiin Node-REDiin asentaa kirjastot, joilla saatiin hoidettua tiedonkeräys ja -lähetysoperaatiot. Tämä onnistui menemällä valikosta ”Manage palette”, joka on ympyröity kuvassa 20.



Kuva 20. Node-RED kirjastojen asennus

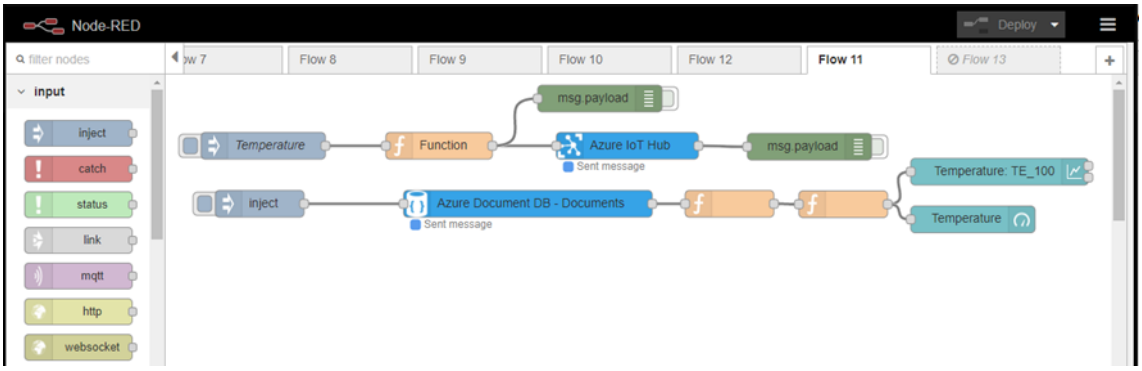
Kirjastoja, joita sovelluksessa tarvittiin, olivat Azure kirjastoja, joita tarvitsi asentaa kaksi kappaletta. Ensimmäinen kirjasto oli IoT Hub-paketti, jota

tarvittiin mittaustietojen lähettämiseen. Tämä kirjasto on kuvassa 21 laatikoituna numerolla 1. Toinen kirjasto, joka tarvittiin asentaa, että mittaus tietoja saatiin luettua tietokannasta, oli DocumentDB kirjasto, joka on kuvassa 21 laatikoituna numerolla 2.



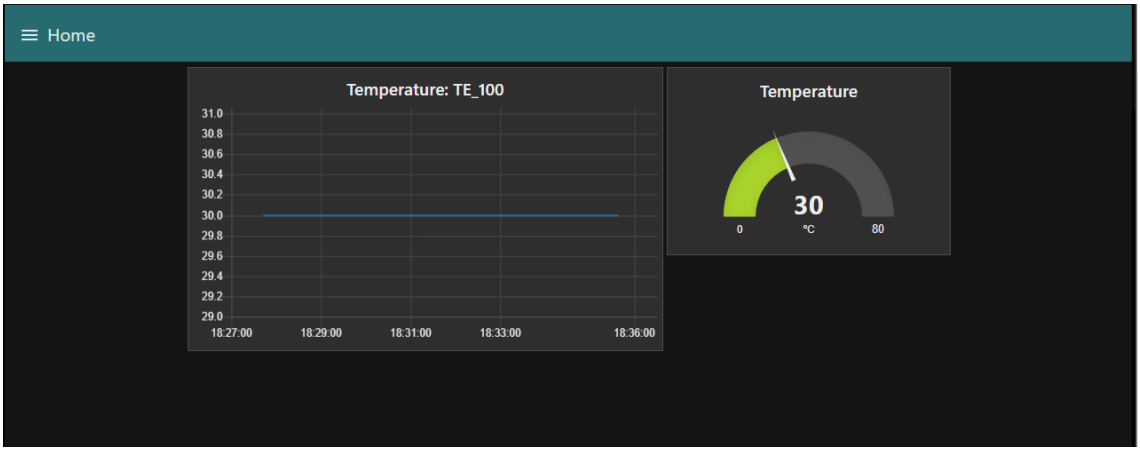
Kuva 21. Node-RED kirjastot

Kun kirjastot oli asennettu, voitiin tehdä kuvan 22 mukainen sovellus, jossa laite lähettää lämpötilatietoa Azuren IoT Hubiin käyttäen hyödyksi asennettua kirjastoa. Samalla tieto voitiin lukea DocumentDBstä käyttäen hyödyksi asennettua kirjastoja. Kehitetty Node-RED sovellus on liitteenä 1.



Kuva 22. Node-RED IoT-sovellus

Tietojen lukemisen jälkeen voitiin tieto muotoilla haluttuun muotoon, jolloin luetusta datasta voitiin muodostaa informaatiosivu, joka on kuvassa 23.



Kuva 23. Node-RED informaatio sivu

9 YHTEENVETO

Tässä työssä käytiin läpi perusteita teollisuuden internetistä ja siihen kuuluvista osa-alueista. Opinnäytetyössä käytiin läpi keskeisiä esineiden internetissä käytettäviä viestintäprotokollia ja tietokantoja. Lisäksi erilaisten pilvipalveluiden tutkiminen oli yksi sovelluksen kehittämistä varten tärkeimmistä osista. Lopuksi materiaalin perusteella luotiin yksinkertainen esimerkki, kuinka hyödynnetään Microsoft Azuren -palveluita esineiden internetin sovelluksessa käyttäen hyödyksi Node-RED kehitysympäristöä.

Keskeinen osa opinnäytetyötä oli IoT-sovellus, jossa luotiin Microsoft Azuren -palveluita käyttäen sovellus, jolla saatiin laitteelta lähetetty tieto tallennettua palvelun tietokantaan käyttäen hyödyksi pilvipalvelun sovelluksia. Ennen esimerkkisovelluksen tekemistä tutustuttiin tarvittavaan teoriaan. Tämän jälkeen tehtiin tarvittavat konfiguroinnit, jotta saatiin kehitysympäristön ja pilvipalvelimen välinen liikennöinti toimimaan halutulla tavalla.

Työn aikana sain peruskäsityksen, mitä esineiden internet tarkoittaa ja minkälaisia asioita tarvitsee ottaa huomioon, että sovelluksesta tulisi toimiva. Sain myös peruskäsityksen siitä, kuinka viestintäprotokollia käytetään datan lähettämiseen ja lukemiseen. Tietokannat olivat minulle uusi käsite aloittaessani tämän opinnäytetyön, mutta opinnäytetyötä edetessä opin paljon uutta asiaa koskien tietokantojen rakenteita ja niiden käytöstä pilvipalvelussa. Myös viestintäprotokollien toiminta eroavaisuudet olivat täysin uutta tietoa minulle. Mutta kokonaisuutena tietotaito, joka tämän työtä toteuttaessa sain, auttaa minua tulevaisuuden haasteissa todella paljon.

Sovelluksen jatkokehitysideana on datan analysointi pilvipalvelussa, joka on yksi kasvavimmista osa-aluista sovelluskehityksen maailmassa. Tässä työssä en päässyt käymään läpi, minkälaisia vaihtoehtoja on jo olemassa datan analysointia varten, johtuen datan vähäisyydestä. Toinen mielenkiintoinen aihe on eri viestintäprotokollien yhdistämistä viestintään laitteen ja pilvipalvelun välille, voisi olla yksi jatkossa tutkittavia asioita.

Suurien kustannuksien vuoksi pilvipalvelua ei vielä otettu hankkeessa käyttöön, vaan sen sijaan otettiin käyttöön paikallinen tietokantaratkaisu, mutta materiaalia voi käyttää hyödyksi, kun tarve ulkopuoliselle tietokannalle tulee jatkossa.

LÄHTEET

AMQP (n.d.). *Advanced Message Queuing Protocol*. Haettu 22.10.2017 osoitteesta https://www.amqp.org/

AWS (n.d.). *Amazon web services*. Haettu 20.10.2017 osoitteesta https://aws.amazon.com/about-aws/

CoAP (n.d.). *Constrained Application Protocol.* Haettu 22.10.2017 osoitteesta http://coap.technology/

Collin, J. & Saarelainen, A. 2016. *Teollinen internet*. Helsinki: Telentum

DocumentDB (2017). *Introduction to Azure Cosmos DB: DocumentDB* API. Haettu 22.11.2017 osoitteesta https://docs.microsoft.com/en-us/azure/cosmos-db/documentdb-introduction

Google a (2017*). Google Cloud Platform*. Kirjoitus blogissa 16.05.2017. Haettu 28.10.2017 osoitteesta https://cloudplatform.googleblog.com/2017/05/introducing-Google-Cloud-IoT-Core-for-securely-connecting-and-managing-IoT-devices-at-scale.html?m=1

Google b (2017*). Google Cloud pricing*. Haettu 28.10.2017 osoitteesta https://cloud.google.com/iot/pricing

Harvey, C (2017). *What is could computing.* Haettu 26.11.2017 osoitteesta https://www.datamation.com/cloud-computing/what-is-cloud-computing.html

Hicklin, J., Shurvinton B. & Beard G. 2015. *The Internet of Things for Dummies.* Chichester, West Sussex, England: John Wiley & Sons, Ltd.

IoT-Agenda (2016). *Internet of Things*. Haettu 23.10.2017 osoitteesta http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

IoT application protocols (2016). Internet of Things Application Protocols. Haettu 23.10.2017 osoitteesta https://www.linkedin.com/pulse/internet-things-iot-application-protocols-narsimhmaswamy-badugu

IoT Hub (n.d.). *IoT Hub pricing*. Haettu 23.10.2017 osoitteesta https://azure.microsoft.com/en-us/pricing/details/iot-hub/

Isode (n.d.). *XMPP protocol*. Haettu 22.10.2017 osoitteesta https://www.isode.com/whitepapers/xmpp.html

Karasiewicz, C. (2013). *Why HTTP is not enough for the Internet of Things*. Kirjoitus blogissa 09.07.2013. Haettu 10.11.2017 osoitteesta https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en

Kobie (2015). *What is internet of things*. Haettu 22.11.2017 osoitteesta https://www.theguardian.com/technology/2015/may/06/what-is-the-internet-of-things-google

Lehtonen, T 2002. *SQL-Opas*. Jyväskylä: Docendo

Microsoft (2017). *Microsoft Streaming Analytics*. Haettu 13.10.2017 osoitteesta https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

Microsoft documentDB (2017). *Microsoft documentDB*. Haettu 13.10.2017 osoitteesta https://docs.microsoft.com/en-us/azure/cosmos-db/documentdb-introduction

Morgan, J (2014). *A Simple Explanation Of 'The Internet Of Things'.* Kirjoitus blogissa 13.11.2014. Haettu 15.11.2017 osoitteesta https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#130c0c3d1d09

MQTT (n.d.). *Message Queue Telemetry Transport*. Haettu 15.10.2017 osoitteesta http://mqtt.org/

MQTT-Broker (2017). *Mqtt-broker for cloud*. Kirjoitus blogissa 12.11.2017. Haettu 22.11.2017 osoitteesta https://pagefault.blog/2017/03/02/using-local-mqtt-broker-for-cloud-and-interprocess-communication/

Node-RED (n.d.). *Flow based programming*. Haettu 22.11.2017 osoitteesta https://nodered.org/about/

Rabbitmq (n.d.) *AMQP concepts*. Haettu 22.10.2017 osoitteesta https://www.rabbitmq.com/tutorials/amqp-concepts.html

Raspberry (n.d.). *Raspberry Pi 3 Model B*. Haettu 22.10.2017 osoitteesta https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

TCP-IP (n.d.). *TCP/IP in The Network Encyclopedia.* Haettu 22.11.2017 osoitteesta http://www.thenetworkencyclopedia.com/entry/tcp-ip/

Uusitalo, I. (2016). *IoT laitteiden turvallisuus*. Kirjoitus blogissa 08.12.2016. Haettu 10.11.2017 osoitteesta https://www.solita.fi/blogit/varmista-iot-laitteiden-tietoturva/

Vieno, J. (2015). *Teollinen internet ja tietoturva*. Kirjoitus blogissa 01.12.2015. Haettu 12.11.2017 osoitteesta https://www.v-tek.fi/teollinen-internet-ja-tietoturva/

XMPP (n.d.). *The most secure messaging standard.* Haettu 05.11.2017 osoitteesta https://xmpp.org/

Liite 1/2

Node-RED ohjelma


```
[
 {
  "id": "ba71d2f7.5e6a4",
  "type": "azureiothub",
  "z": "81fe6060.cfabb",
  "name": "Azure IoT Hub",
  "protocol": "mqtt",
  "x": 520,
  "y": 80,
  "wires": [
   [
    "a615034.599e4"
   ]
  ]
 },
 {
  "id": "a615034.599e4",
  "type": "debug",
  "z": "81fe6060.cfabb",
  "name": "",
  "active": false,
  "console": "false",
  "complete": "payload",
  "x": 725,
  "y": 81,
  "wires": []
 },
 {
  "id": "37b6e6a5.874cea",
  "type": "function",
  "z": "81fe6060.cfabb",
  "name": "Function",
  "func": "value= Math.random()*(50-50)+30;\nrnd= Math.round(value*100)/100;\nmsg1 = '{ \"deviceId\": \"TE_100\",\"key\": \"kW2F2zTcuiC3E2l0SX4/rNpuD+OmS23GSLLmxIH35Aw=\",\"protocol\": \"mqtt\",\"data\": { \"time\":'+Date.now()+',\"value\": ' + rnd + ',\"unit\":\"celsius\"}}';\n\nnewMsg = { payload: msg1 };\nreturn newMsg;\n",
  "outputs": 1,
  "noerr": 0,
  "x": 308,
  "y": 80,
  "wires": [
   [
    "ba71d2f7.5e6a4",
```

Liite 1/2


```
"86f9cf59.7209a"
     ]
    ]
  },
  {
    "id": "36b1e39b.4224ec",
    "type": "inject",
    "z": "81fe6060.cfabb",
    "name": "Temperature",
    "topic": "TE_100",
    "payload": "",
    "payloadType": "num",
    "repeat": "",
    "crontab": "",
    "once": false,
    "x": 110,
    "y": 80,
    "wires": [
      [
        "37b6e6a5.874cea"
      ]
    ]
  },
  {
    "id": "a1ec3577.f695a8",
    "type": "Documents",
    "z": "81fe6060.cfabb",
    "name": "Azure Document DB - Documents",
    "x": 408,
    "y": 145,
    "wires": [
      [
        "cbe7da64.a03d18"
      ]
    ]
  },
  {
    "id": "63d637c8.1fc1f8",
    "type": "inject",
    "z": "81fe6060.cfabb",
    "name": "",
    "topic": "",
    "payload":                    "{\"dbname\":\"ToDoList\",\"collName\":\"Items\",\"ac-
tion\":\"Q\",\"query\":\"SELECT c.timestamp,c.deviceid ,c.val FROM root c where c.de-
viceid = \\\"TE_100\\\" and c.val != null\"}",
    "payloadType": "json",
    "repeat": "",
```

```
"crontab": "",
   "once": false,
   "x": 106,
   "y": 145,
   "wires": [
    [
     "a1ec3577.f695a8"
    ]
   ]
  },
  {
   "id": "86f9cf59.7209a",
   "type": "debug",
   "z": "81fe6060.cfabb",
   "name": "",
   "active": false,
   "console": "false",
   "complete": "false",
   "x": 489,
   "y": 27,
   "wires": []
  },
  {
   "id": "cbe7da64.a03d18",
   "type": "function",
   "z": "81fe6060.cfabb",
   "name": "",
   "func": "var st = msg;\nmsg = {};\nmsg.payload = JSON.parse(st.substring(19,
st.length));\nreturn msg;",
   "outputs": 1,
   "noerr": 0,
   "x": 635,
   "y": 145,
   "wires": [
    [
     "a19ce9eb.233c98"
    ]
   ]
  },
  {
   "id": "a19ce9eb.233c98",
   "type": "function",
   "z": "81fe6060.cfabb",
   "name": "",
   "func": "//msg = msg.substring(19,msg.length);\n//msg.payload =
JSON.parse(msg.payload)\nmsg1 = msg.payload;\n///msg2 = msg.payload;\nvar i;\nvar
```

Liite 1/4


```
ts = msg.payload.length;\nfor (i =0; i < msg1.length; i++)\n{\n      msg.payload= msg1[i].val;\n   //msg2.payload= msg2[i].time;\n}\nreturn msg;",
   "outputs": 1,
   "noerr": 0,
   "x": 763,
   "y": 145.5,
   "wires": [
    [
     "7a3e5e0f.d47b9",
     "ad28c634.324c88"
    ]
   ]
  },
  {
   "id": "7a3e5e0f.d47b9",
   "type": "ui_chart",
   "z": "81fe6060.cfabb",
   "name": "",
   "group": "253ab448.cfc09c",
   "order": 0,
   "width": 0,
   "height": 0,
   "label": "Temperature: TE_100",
   "chartType": "line",
   "legend": "false",
   "xformat": "HH:mm:ss",
   "interpolate": "linear",
   "nodata": "Temperature",
   "dot": false,
   "ymin": "",
   "ymax": "",
   "removeOlder": "24",
   "removeOlderPoints": "",
   "removeOlderUnit": "3600",
   "cutout": 0,
   "useOneColor": false,
   "colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",
    "#2ca02c",
    "#98df8a",
    "#d62728",
    "#ff9896",
    "#9467bd",
    "#c5b0d5"
   ],
```

```
"x": 939,
    "y": 109,
    "wires": [
     [],
     []
    ]
   },
   {
    "id": "ad28c634.324c88",
    "type": "ui_gauge",
    "z": "81fe6060.cfabb",
    "name": "Temperature",
    "group": "908cc36.d4fb94",
    "order": 0,
    "width": 0,
    "height": 0,
    "gtype": "gage",
    "title": "Temperature",
    "label": "ºC",
    "format": "{{value}}",
    "min": 0,
    "max": "80",
    "colors": [
     "#00b500",
     "#e6e600",
     "#ca3838"
    ],
    "seg1": "",
    "seg2": "",
    "x": 909,
    "y": 162.5,
    "wires": []
   },
   {
    "id": "253ab448.cfc09c",
    "type": "ui_group",
    "z": "",
    "name": "Chart",
    "tab": "d3e0bfe5.9967b",
    "disp": false,
    "width": "10"
   },
   {
    "id": "908cc36.d4fb94",
    "type": "ui_group",
    "z": "",
    "name": "temperature",
```

"tab": "d3e0bfe5.9967b",
    "disp": false,
    "width": "6"
  },
  {
    "id": "d3e0bfe5.9967b",
    "type": "ui_tab",
    "z": "",
    "name": "Home",
    "icon": "dashboard",
    "order": 1
  }
]